


普通高等教育"十一五"国家级规划教材

高等院校信息与通信工程系列教材

# 网络安全协议理论与技术

范明钰　王光卫　编著

清华大学出版社

高等院校信息与通信工程系列教材
普通高等教育“十一五”国家级规划教材

# 网络安全协议理论与技术

范明钰　王光卫　编著

清华大学出版社
北　京

## 内 容 简 介

本书从基本概念入手,通过 Internet 协议的实际例子,建立网络协议的概念,分析了 Internet 协议不安全的原因,介绍了安全协议的密码学基础,分析了安全协议与密码学的关系,介绍了利用不同的密码算法建立安全信道。从第 4 章开始,介绍基本的安全协议、抗攻击的安全协议和实际使用的安全协议。附录中介绍了最新的几类密码算法。每章都附有重点和难点分析,并附有习题与思考题。

全书共分为三个部分:第一部分介绍基本概念和 Internet 中的协议(第 1 章和第 2 章)。第二部分介绍安全协议,分为三个内容、安全协议的密码学基础(第 3 章)、基本安全协议(第 4 章)和抗攻击的安全协议(第 5 章)。第三部分介绍实际使用的安全协议(第 6 章)。这三个部分基本上是关联的,既可以从概念入手讲解,也可以先从实际例子开始最后得到理性的知识。

本书可供工科类计算机、电子信息、通信等相关学科的本科学生和研究生使用。



**图书在版编目(CIP)数据**

网络安全协议理论与技术/范明钰,王光卫编著. —北京:清华大学出版社,2009.2
(高等院校信息与通信工程系列教材)
ISBN 978-7-302-19300-5

Ⅰ. 网…　Ⅱ. ①范…②王…　Ⅲ. 计算机网络—安全技术—通信协议—高等学校—教材
Ⅳ. TP393.08

中国版本图书馆 CIP 数据核字(2009)第 006103 号

**责任编辑:** 陈国新
**责任校对:** 白　蕾
**责任印制:**

**出版发行:** 清华大学出版社　　**地　址:** 北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　编:** 100084
**社 总 机:** 010-62770175　　**邮　购:** 010-62786544
**投稿与读者服务:** 010-62776969,c-service@tup.tsinghua.edu.cn
**质 量 反 馈:** 010-62772015,zhiliang@tup.tsinghua.edu.cn
**印 刷 者:**
**装 订 者:**
**经　销:** 全国新华书店
**开　本:** 185×260　　**印　张:** 15　　**字　数:** 347 千字
**版　次:** 2009 年 2 月第 1 版　　**印　次:** 2009 年 2 月第1 次印刷
**印　数:** 1～　000
**定　价:** 　.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103　　产品编号:

# 出 版 说 明

信息与通信工程学科是信息科学与技术的重要组成部分。改革开放以来,我国在发展通信系统与信息系统方面取得了长足的进步,形成了巨大的产业与市场,如我国的电话网络规模已位居世界首位,同时该领域的一些分支学科出现了为国际认可的技术创新,得到了迅猛的发展。为满足国家对高层次人才的迫切需求,当前国内大量高等学校设有信息与通信工程学科的院系或专业,培养大量的本科生与研究生。为适应学科知识不断更新的发展态势,他们迫切需要内容新颖又符合教改要求的教材和教学参考书。此外,大量的科研人员与工程技术人员也迫切需要学习、了解、掌握信息与通信工程学科领域的基础理论与较为系统的前沿专业知识。为了满足这些读者对高质量图书的渴求,清华大学出版社组织国内信息与通信工程国家级重点学科的教学与科研骨干以及本领域的一些知名学者、学术带头人编写了这套高等院校信息与通信工程系列教材。

该套教材以本科电子信息工程、通信工程专业的专业必修课程教材为主,同时包含一些反映学科发展前沿的本科选修课程教材和研究生教学用书。为了保证教材的出版质量,清华大学出版社不仅约请国内一流专家参与了丛书的选题规划,而且每本书在出版前都组织全国重点高校的骨干教师对作者的编写大纲和书稿进行了认真审核。

祝愿《高等院校信息与通信工程系列教材》为我国培养与造就信息与通信工程领域的高素质科技人才,推动信息科学的发展与进步做出贡献。

北京邮电大学<br>陈俊亮<br>2004 年 9 月

# 前　　言

本书的编写经历了近四年，主要作为信息安全类本科生和研究生的教学参考书。

全书从基本概念入手，通过 Internet 协议的实际例子，建立网络协议的概念，分析了 Internet 协议不安全的原因，介绍了安全协议的密码学基础，分析了安全协议与密码学的关系，介绍了利用不同的密码算法建立安全信道。从第 4 章开始，介绍基本的安全协议、抗攻击的安全协议和实际使用的安全协议。附录中介绍了最新的几类密码算法。每章都附有重点和难点分析，并附有习题与思考题。

本书共分为三个部分。

第一部分：基本概念和 Internet 中的协议（第 1 章和第 2 章）。

第二部分：安全协议，分为三个内容：安全协议的密码学基础（第 3 章）、基本安全协议（第 4 章）、抗攻击的安全协议（第 5 章）。

第三部分：实际使用的安全协议（第 6 章）。

本书的三个部分基本上是关联的，既可以从概念入手讲解，也可以先从实际例子开始最后得到理性的知识。

参与本教材编写的主要人员有：王庆先博士、朱大勇博士；实验室学生沈丹、丁旭阳、张涛，其中第 4 章和第 5 章的绝大部分插图以及附录是沈丹同学编制的。谨在这里向他们表示诚挚的谢意。

编　者

2008 年 10 月

# 目　录

# 第1章　基本概念

本章是网络安全协议中关于网络、安全及密码学的基础知识。

本章分5个小节，第1.1节介绍网络基础以及网络协议的概念；第1.2节介绍网络安全的概念；第1.3节介绍网络的协议；第1.4节介绍网络协议面临的威胁；第1.5节是本章重点和难点分析。

## 1.1　网络基础及网络协议的概念

简单地说，网络是由两台以上计算机借助于协议连在一起组成的“计算机群”，再加上相应“通信设备”组成的综合系统。

早期的计算机应用模式是单机，其发展过程有小型机、中型机、大型机。单台计算机能干很多事情。虽然计算机的速度越来越快、性能越来越高、容量越来越大，但还是存在一些美中不足。比如办公室为每个人都配备了一台最新式计算机，但是打印机的配备却成了问题。如果只为一台或者几台计算机配备打印机，那些没有配备打印机的人打印时就需要把文件用磁盘复制到有打印机的计算机上去打印，不仅麻烦，而且也耽误别人的时间。另一方面，如果给所有计算机都配备打印机，它们多数情况下是处于闲置状态，很明显这是一种浪费。如果只给一台或几台计算机配备打印机，而其他所有计算机都可以利用这些打印机，并且相互之间不影响工作，这就是资源共享。

可以在网络上共享的资源除了打印机之外，还有硬盘、光盘、绘图仪、扫描仪以及各类软件、文本和各种信息资源等。在网络中共享资源既节省了大量的投资和开支，又便于集中管理。

利用网络可以进行信息交换和信息的集中与分散处理，比如说一家公司，有生产部、仓储部、市场部、财务部等很多部门和分公司。这些部门和分公司在地理位置上并不在一起。但是作为一个现代化的大公司，各个业务部门需要随时知道其他部门的各种数据：分散的销售数据需要及时集中起来配合仓储部的库存和生产部的生产，分散的财务数据也需要随时送到财务部集中处理以配合公司的整体行动。诸如此类，称为信息交换和信息的集中与分散处理。这些都需要依托网络才能做到。

计算机网络并不是随着计算机的出现而出现的，而是随着社会对资源共享和信息交换与及时传递的迫切需要而发展起来的。它是现代计算机技术和通信技术密切结合的产物。说得准确一些，计算机网络就是利用通信设备和通信线路，把位于不同地点的计算机等设备相互联起来，用相应的协议软件实现资源共享和信息交换的系统。

早期的网络是一个单位的几台计算机用一根电缆串在一起，实现局部资源共享和信

息交换。今天的网络，把世界上百个国家的大大小小几千万台计算机连为一体，形成硕大无比像蜘蛛网一样的“怪物”，在全世界范围内实现全方位的资源共享和信息交换。这就是 Internet，也称为国际互联网或因特网。对于一个单位来说，只要把这个单位网络的对外连线往 Internet 一搭，网络性质就从根本上改变了，其外延与内涵都产生了根本的变化。

网络带来的好处主要体现在资源共享、信息交换与及时传递两个方面。就拿资源共享来说吧，一个办公室或者几个办公室只安装一台打印机而不耽误工作；一个公司或者图书馆只购买一份昂贵的软件，公司里所有人都可以随意使用；火车站或者航空公司售票处，把票务信息汇总后放在网上，任何人都可以随时在网上查阅，知道某一次列车或者航班还有多少张票。诸如此类，既节约资金，又减少重复劳动。

在网络中进行信息交换与及时传递好处则更大。因为有了计算机网络，《人民日报》就能在北京制完版后几分钟内，将版样传送到全国各地，甚至国外的印制点。这样，在早晨 6 点多钟便可以从报上知道报纸印制前半小时发生的新闻；也是因为有了网络，花都的农民在家中便可以把鲜花推销到世界各国；韶关的孩子坐在家中就可以上广州师范附中的网校，接受全国特级教师的课外辅导。可以这么说，正是因为可以通过网络进行远距离的信息交换和及时传递，网络改变了时空，人与人之间的距离变近了，地球变小了，信息变多了。

### 1.1.1 网络的构成和分类

计算机网络是计算机技术和数据通信技术紧密结合的产物。所谓计算机网络，通俗地讲，就是将地理位置不同的多个计算机系统通过通信设备和线路连接起来，以功能完善的网络软件（在协议控制下）实现网络中资源共享和数据交换的系统，见图 1-1。




图 1-1 网络的构成

一个用计算机联网的通信系统一般由 6 个部分组成。

（1）信息（message）：包括文字、声音、图像等数据。

（2）发送设备：又称“主机”（host）——各种信息处理设备（计算机等）。

（3）接收设备：同发送设备。

（4）通信设备：负责主机间的通信控制和通信处理。

（5）传输媒介：各种电缆、光纤、无线电波等。

（6）通信协议：通信规则（无协议的两台设备可以连接但无法通信，如同讲不同语言的两人无法对讲）。

网络可分为资源子网和通信子网两部分，见图 1-2。

其中，资源子网包括硬件资源（主机、终端、I/O 设备等）、软件资源、数据资源等，负责全网数据处理业务，向网络用户提供各种网络资源和网络服务；通信子网包括传输介质

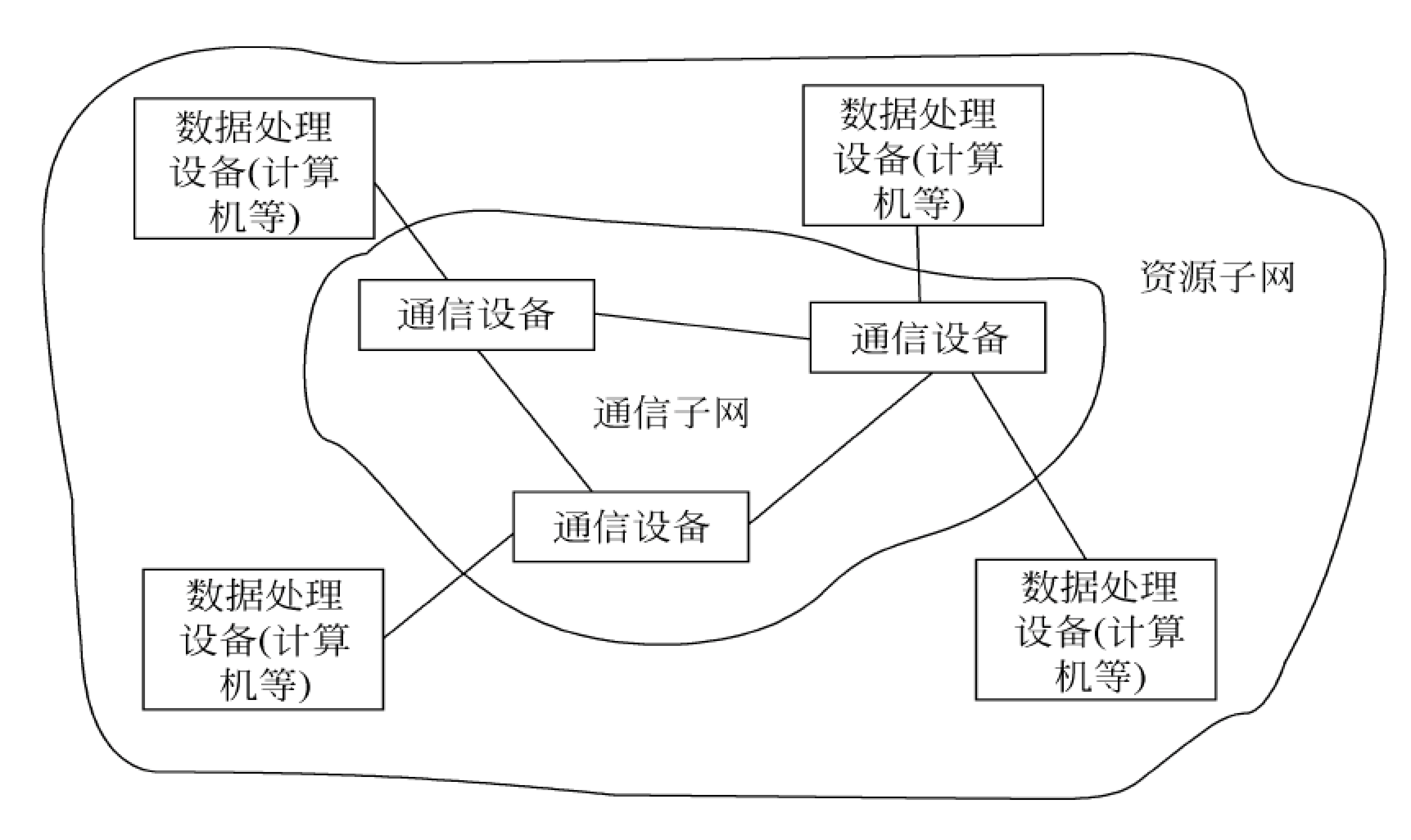



图 1-2　资源子网和通信子网

(电缆、光纤、无线电波等)、通信设备(交换机等)，承担全网的数据传输、转接、加工和变换等通信处理工作。

按网络的规模和地理位置，网络可分为如下几种。

(1) 局域网(local area network，LAN)：一般在小于 10km 的范围区域内，通常采用有线的方式连接起来。局域网通常用于一个单位、一座大楼或相应楼群之间，也特别适合于一个地域跨度不大的企业建立内部网，即 Intranet。

(2) 园区网：介于局域网和广域网之间的网络。

(3) 城域网(metropolitan area network，MAN)：规模局限在一座城市的范围内，10～100km 的区域。

(4) 广域网(wide area network，WAN)：网络跨越国界、洲界，甚至全球范围。Internet 是著名的广域网。

按网络权限关系，网络可分为内部网(intranet)和外部网(extranet)。

按照拓扑结构，网络可以分为总线型、星形、环形和网格网(全连网格网与不全连网格网)，参见图 1-3 和图 1-4。

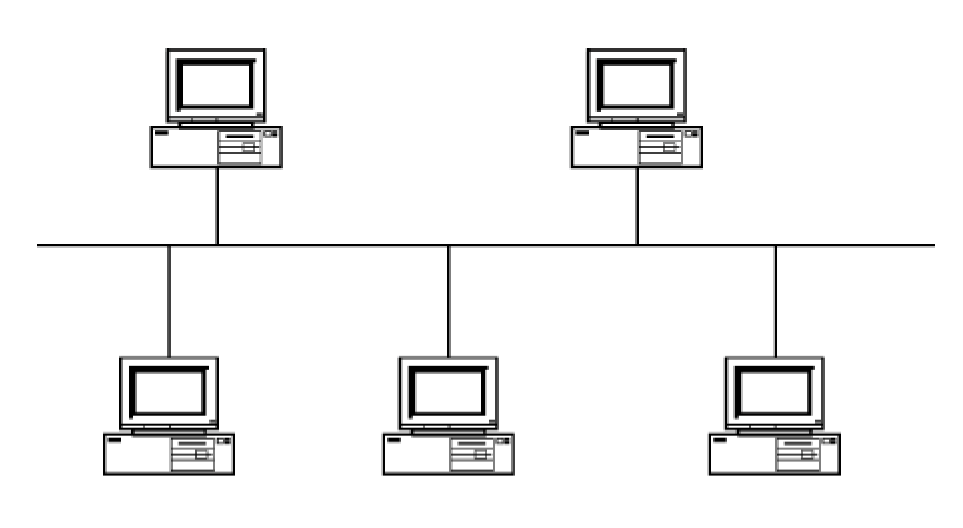


图 1-3　总线型网络拓扑

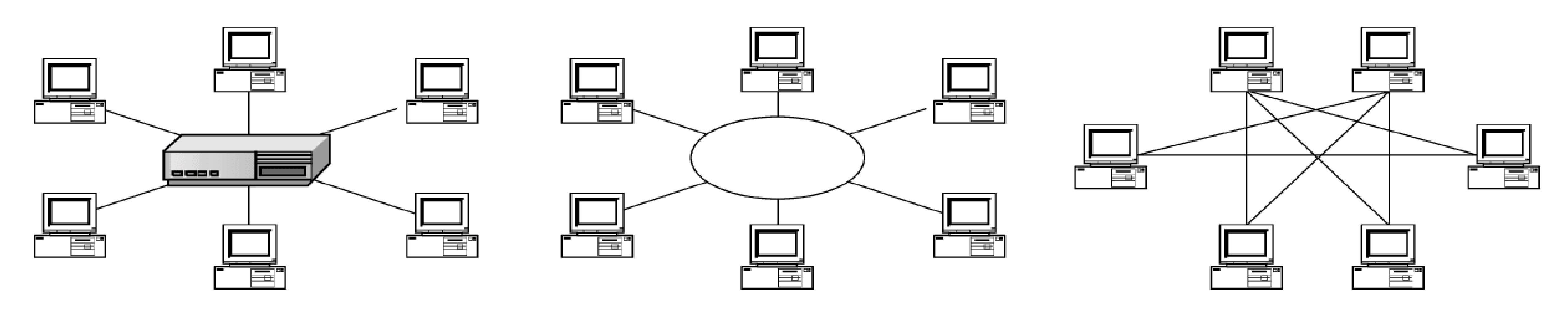


图 1-4　星形网络拓扑、环形以太网拓扑和网格网拓扑

按传输介质,网络可以分为如下几种。

(1) 有线网:采用同轴电缆或双绞线来连接的计算机网络。因速度有限,技术落后,已被淘汰。

双绞线网是目前最常见的联网方式。它价格便宜,安装方便。因距离短,适于局域网内。

(2) 光纤网:光纤网也是有线网的一种,但由于其特殊性而单独列出。光纤网采用光导纤维做传输介质,光纤传输距离长,传输率高,可达数千兆比特每秒,甚至更高,抗干扰能力强,不会受到电子监听设备的监听,是高安全性网络的理想选择。已被广泛应用。

(3) 无线网:采用空间做传输介质,用电磁波作为载体来传输数据。由于联网方式灵活方便,是一种很有前途的联网方式。

## 1.1.2 网络的发展

计算机网络产生于 20 世纪 60 年代,如前所述,其发展动力主要有资源共享的需求、大型项目的合作,以及人与人之间的沟通需要。

按体系结构的发展来分,网络的发展过程大致可以分为以主机为中心的联机终端系统、以通信子网为中心的主机互联,以及具有层次化体系结构的标准化网络三个阶段。

### 1. 以主机为中心的联机终端系统

这种联机系统是早期网络的雏形,其特征主要是共享主机软硬件资源,其构成可分为单台主机(担负计算和通信任务)和多台终端(担负与用户的交互任务)。这种网络中,连接方式主要是本地或远程连接,如图 1-5 所示。

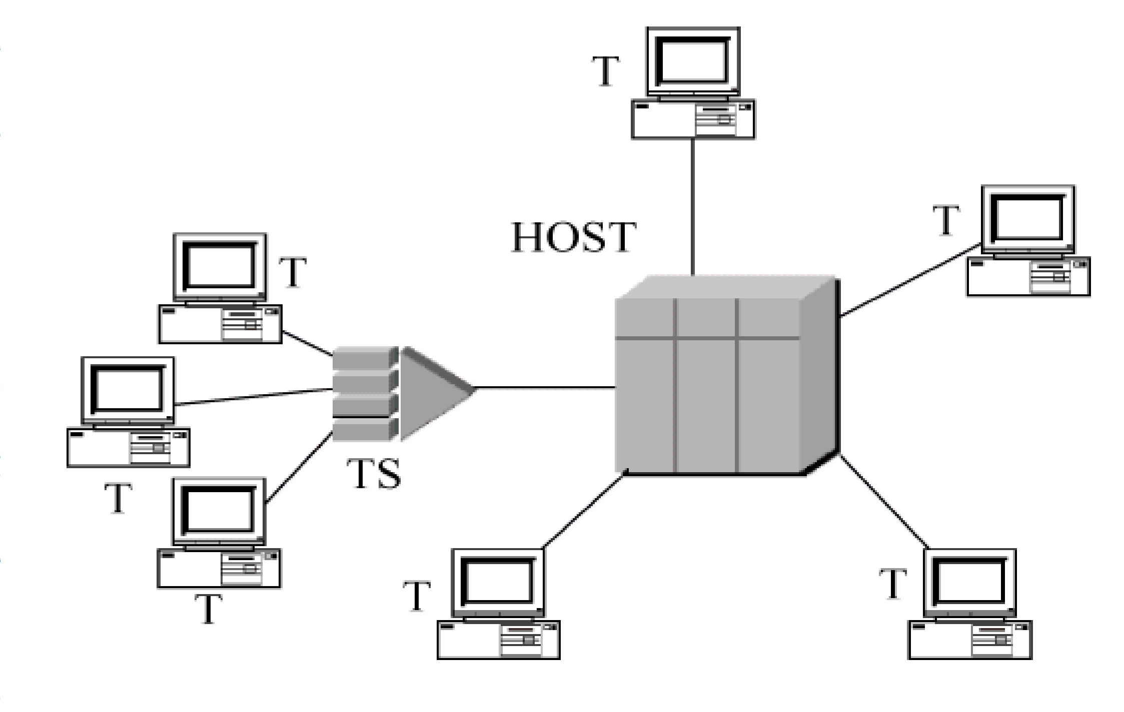



图 1-5 以主机为中心的联机终端系统

这种网络的例子有飞机订票系统,其中 HOST 为航空公司,终端为各订票点,采用的通信线路一般为电话线路。这种网络的缺点主要是主机负荷重,既要完成数据处理还要进行通信,此外线路利用率也低。对这种网络的改进方法是,终端集中器(集线器)加上主机的前端处理机,使通信任务与处理任务分离。

### 2. 以通信子网为中心的主机互联

这种网络的特征是多个终端联机系统的互联,形成以多主机为中心的网络,网络结构从“主机-终端”转变为“主机-主机”,如图 1-6 所示。

### 3. 具有层次化体系结构的标准化网络的演变

主机-主机网络的演变如下。

(1) 演变阶段 1:通信任务从主机中分离,由通信控制处理机 CCP 完成,CCP 是处理主机之间通信任务的专用计算机,见图 1-7。

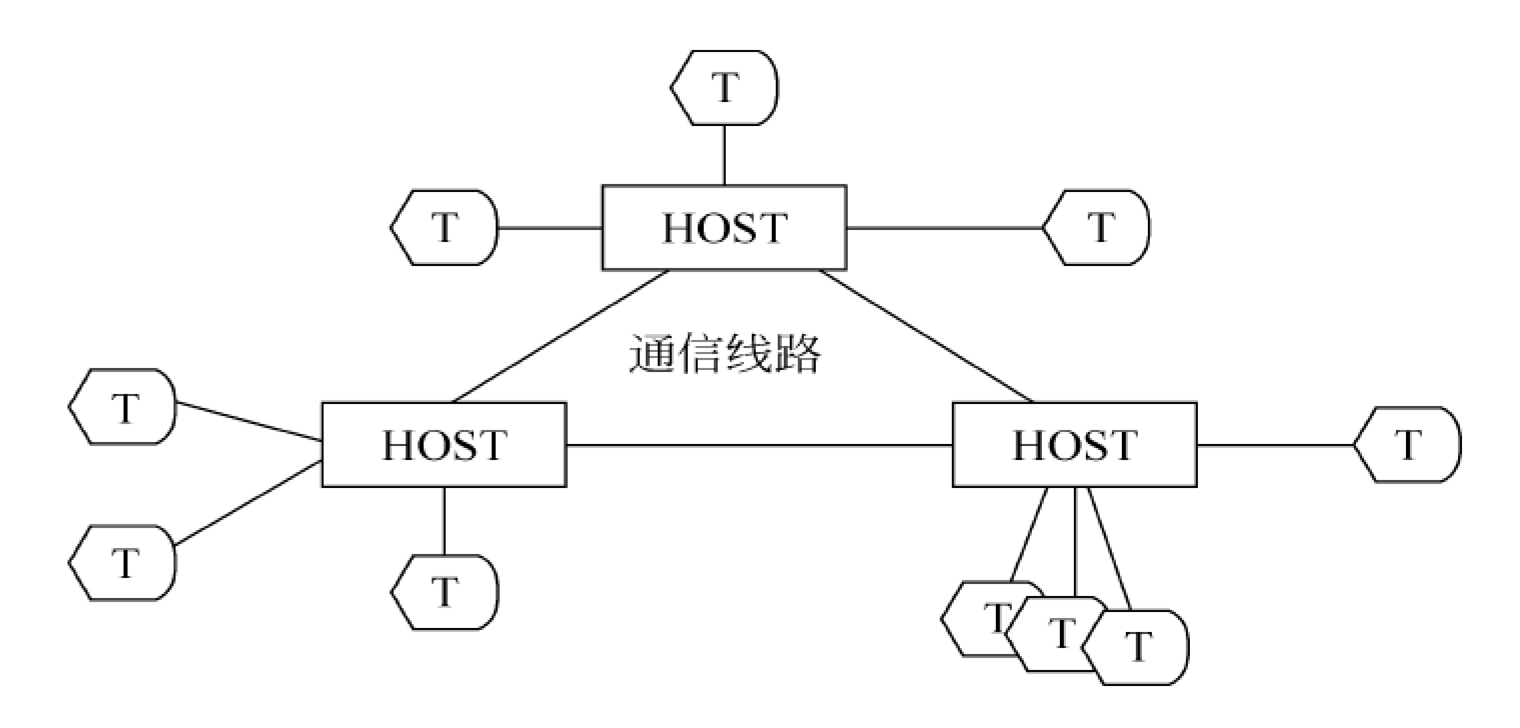



图 1-6　以通信子网为中心的主机互联

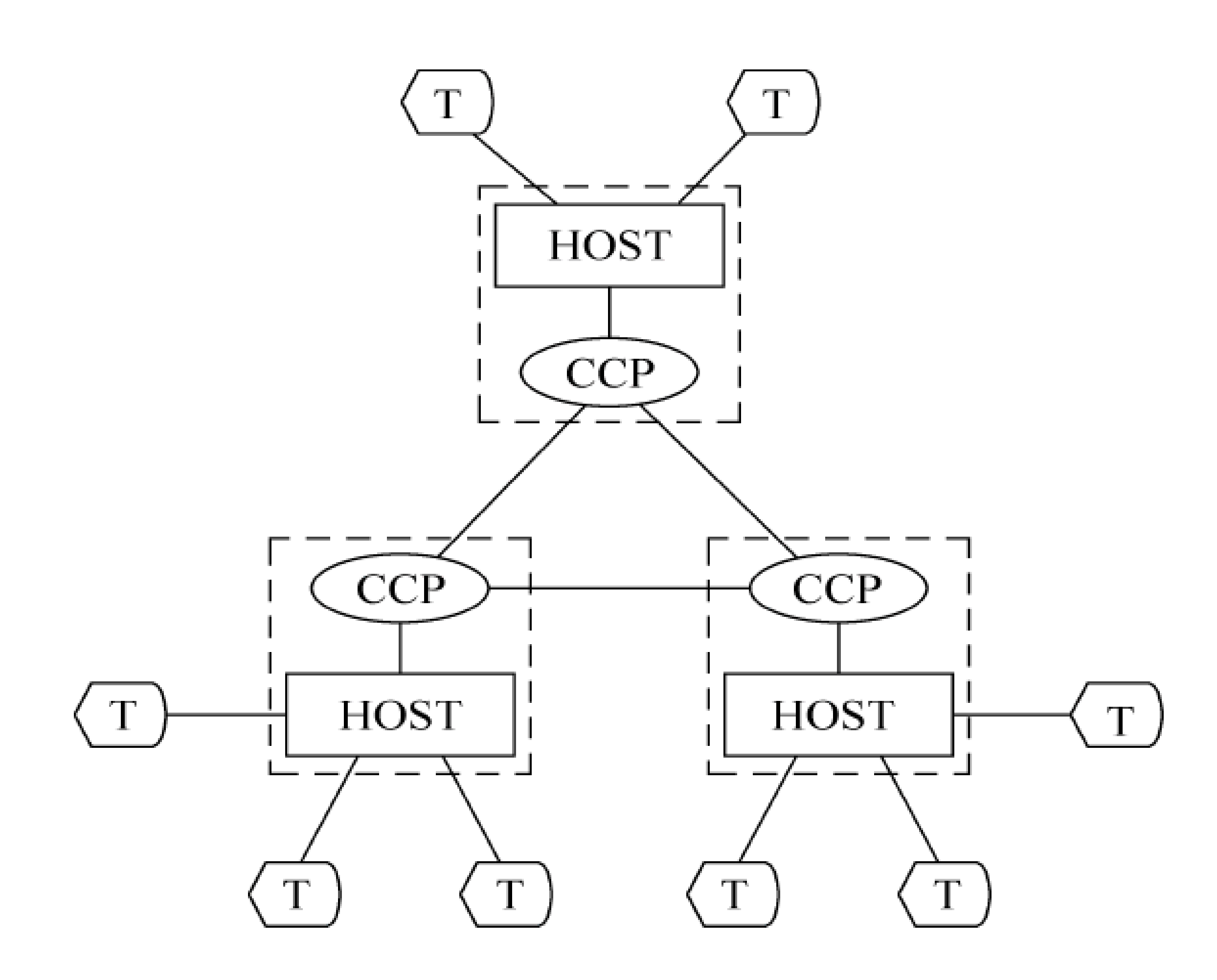



图 1-7　主机-主机网络的演变第一阶段

(2) 两层网络概念的出现：由 CCP 组成的传输网络——通信子网，为主机提供信息传输服务；建立在通信子网基础上的主机集合——资源子网，提供计算资源，见图 1-8。在两层网络的通信子网上可有多个资源子网，共享通信子网的服务。

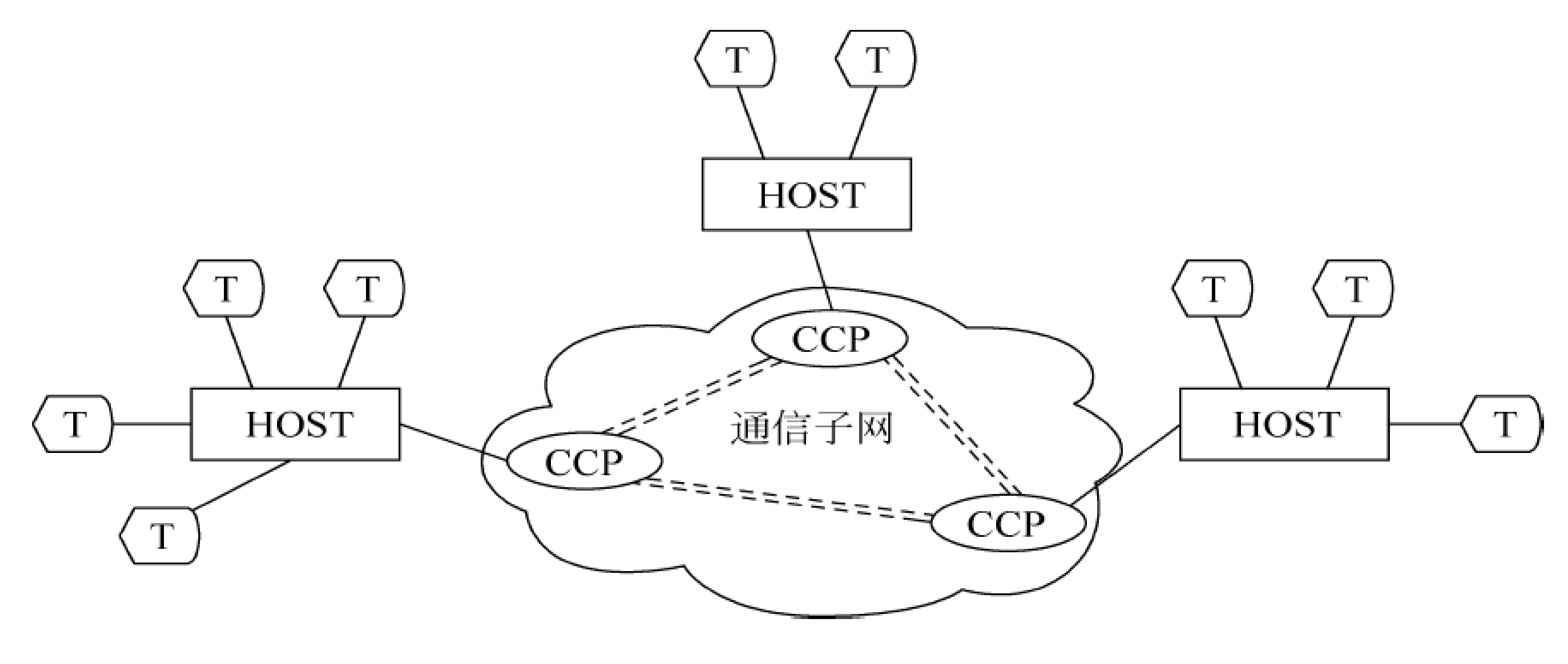



图 1-8　两层网络

(3) 使用公用数据网络：通信子网规模逐渐扩大，从私有网络拓展到社会公用网络，利用公用数据通信网(如 PSTN、X. 25 等)实现，见图 1-9。

一台计算机连入网络以后，具有共享资源、提高可靠性、分担负荷和实现实时管理等优点。网络具有以下几个特点：

(1) 开放式的网络体系结构，使不同软硬件环境、不同网络协议的网可以互联，真正达到资源共享、数据通信和分布处理的目标。

(2) 向高性能发展。追求高速、高可靠和高安全性，采用多媒体技术，提供文本、声音、图像等综合性服务。

(3) 网络的智能化，多方面提高网络的性能和综合的多功能服务，更加合理地进行各种网络业务的管理，真正以分布和开放的形式向用户提供服务。

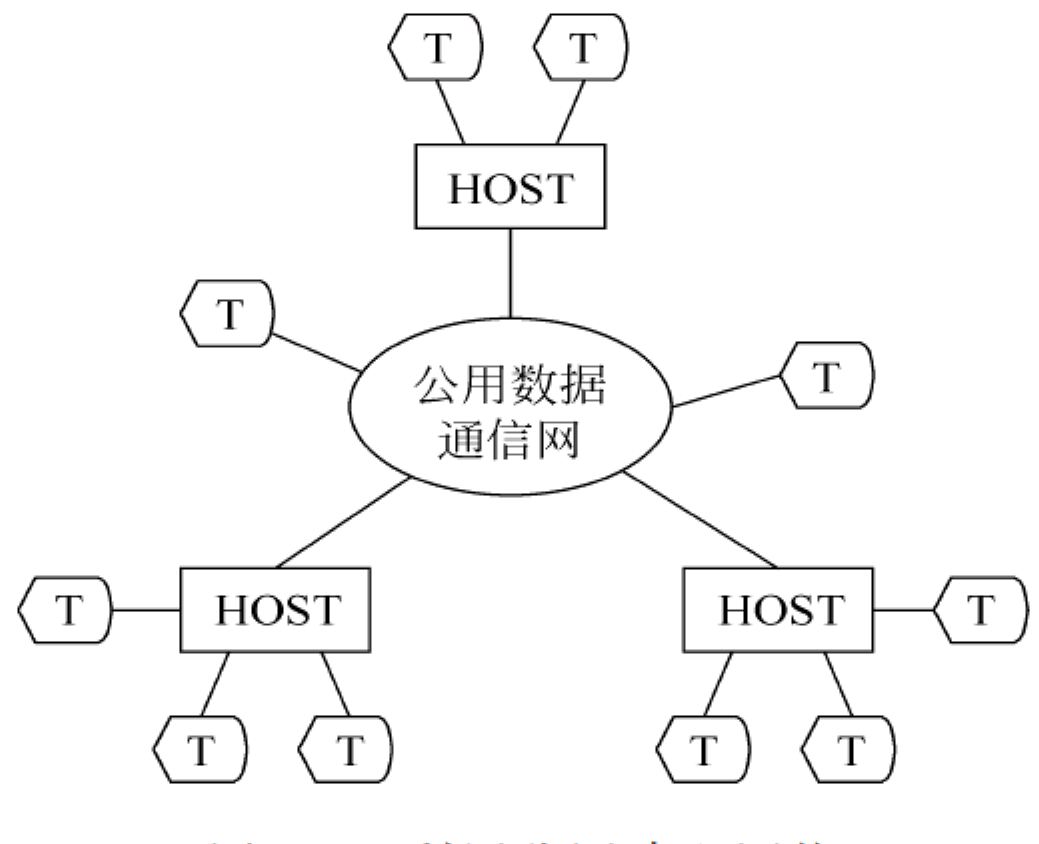
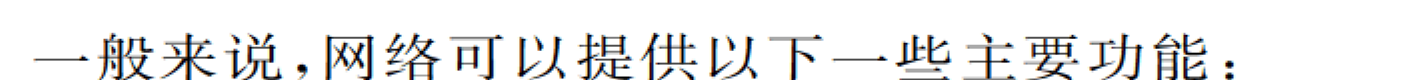



图 1-9 利用公网建立网络


一般来说，网络可以提供以下一些主要功能：

(1) 资源共享。

(2) 信息传输与集中处理。

(3) 均衡负荷与分布处理。

(4) 综合信息服务。

目前，网络还处于迅速发展的阶段。网络技术的不断更新，进一步扩大了网络的应用范围。除了前面提到的资源共享和信息传输等基本功能外，网络还具有以下几个主要方面的应用。

(1) 远程登录。允许一个地点的用户与另一个地点的计算机上运行的应用程序进行交互对话。

(2) 电子邮件。网络可以作为通信媒介，用户可以在自己的计算机上把电子邮件(e-mail)发送到世界各地，这些邮件中可以包括文字、声音、图形、图像等信息。

(3) 电子数据交换。电子数据交换(EDI)是网络在商业中的一种重要的应用形式。它以共同认可的数据格式，在贸易伙伴的计算机之间传输数据，代替了传统的贸易单据，从而节省了大量人力和财力，提高了效率。

(4) 联机会议。利用网络，人们可以通过个人计算机参加会议讨论。联机会议除了可以使用文字外，还可以传送声音、图形和视频。总之，网络的应用范围非常广泛，它已经渗透到国民经济以及人们日常生活的各个方面。

## 1.2 网络安全的概念

随着计算机信息系统应用的深入，计算机信息系统逐步从单机向局域网、广域网发展，特别是 Internet 的迅速发展，计算机信息系统安全面临新的、更严峻的挑战。

构成计算机信息系统基础的计算机操作系统和网络的千差万别，实现计算机连接的多种网络拓扑结构的混合，所采用介质的多样性，信息的集中处理或分布处理等多种形式，这些都大大增加了计算机信息系统安全问题解决的难度。因此，计算机信息

系统安全不再是系统内某个元素或某几个元素的安全，而是系统整体的安全；不再是一个单纯而简单的问题，而是一个系统工程。从技术角度看，计算机系统安全包括计算机安全、网络安全和信息安全。其中信息安全是主线，它贯穿在计算机安全和网络安全之中。

### 1.2.1　网络安全的含义

网络安全的具体含义随着“角色”的不同而变化。具体有以下几个方面：

(1) 从用户（个人、企业等）的角度来说，希望涉及个人隐私或商业利益的信息在网络上传输时受到机密性、完整性和真实性的保护，避免其他人或对手利用窃听、冒充、篡改、抵赖等手段侵犯用户的利益和隐私，同时也避免其他用户的非授权访问和破坏。

(2) 从网络运行和管理者角度来说，希望对本地网络信息的访问、读写等操作受到保护和控制，避免出现“陷阱”、病毒、非法存取、拒绝服务、网络资源非法占用和非法控制等威胁，制止和防御网络黑客的攻击。

(3) 对安全保密部门来说，希望对非法的、有害的或涉及国家机密的信息进行过滤和防堵，避免机要信息泄露，避免对社会产生危害，对国家造成巨大损失。

(4) 从社会教育和意识形态角度来说，网络上不健康的内容会对社会的稳定和人类的发展造成损害，必须对其进行控制。

(5) 但是从本质上来说，网络安全就是网络上信息的安全，是指网络系统的硬件、软件及其系统中的数据受到保护，不因偶然的或者恶意的原因而遭到破坏、更改、泄露，系统连续、可靠、正常地运行，网络服务不中断。广义来说，凡是涉及网络上信息的保密性、完整性、可用性、真实性和可控性的相关技术和理论都是网络安全所要研究的领域。网络安全涉及的内容既有技术方面的问题，也有管理方面的问题，两方面相互补充，缺一不可。技术方面主要侧重于防范外部非法用户的攻击，管理方面则侧重于内部人为因素的管理。如何更有效地保护重要的信息数据、提高网络系统的安全性已经成为所有网络应用必须考虑和必须解决的一个重要问题。

### 1.2.2　不同环境和应用中的网络安全

运行系统安全：保证信息处理和传输系统的安全。侧重于保证系统正常运行，避免因为系统的崩溃和损坏而对系统存储、处理和传输的信息造成破坏和损失，避免由于电磁泄漏，产生信息泄露，干扰他人或受他人干扰。

网络系统信息的安全：包括用户密码认证，用户存取权限控制，数据存取权限和方式控制，安全审计，安全问题跟踪，计算机病毒防治，数据加密等。

网络信息传播安全：信息传播过程和后果的安全，包括信息过滤等。侧重于防止和控制非法、有害的信息进行传播的后果，避免公用网络上大量自由传输的信息失控。

网络信息内容的安全：侧重于信息的保密性、真实性和完整性。避免攻击者利用系统的安全漏洞进行窃听、冒充、诈骗等有损于合法用户的行为，本质上是保护用户的利益和隐私。

### 1.2.3 网络安全的重要性

随着计算机网络的广泛使用和网络之间信息传输量的急剧增长，一些机构和部门在得益于网络加快业务运作的同时，其数据也有可能遭受破坏，或被删除或被复制，自身利益受到严重威胁。

根据国家计算机病毒应急处理中心和计算机病毒防治产品检验中心的调查，我国计算机病毒感染率在连续两年呈下降趋势后，2007 年又出现较大反弹达到 91.47%，见图 1-10。

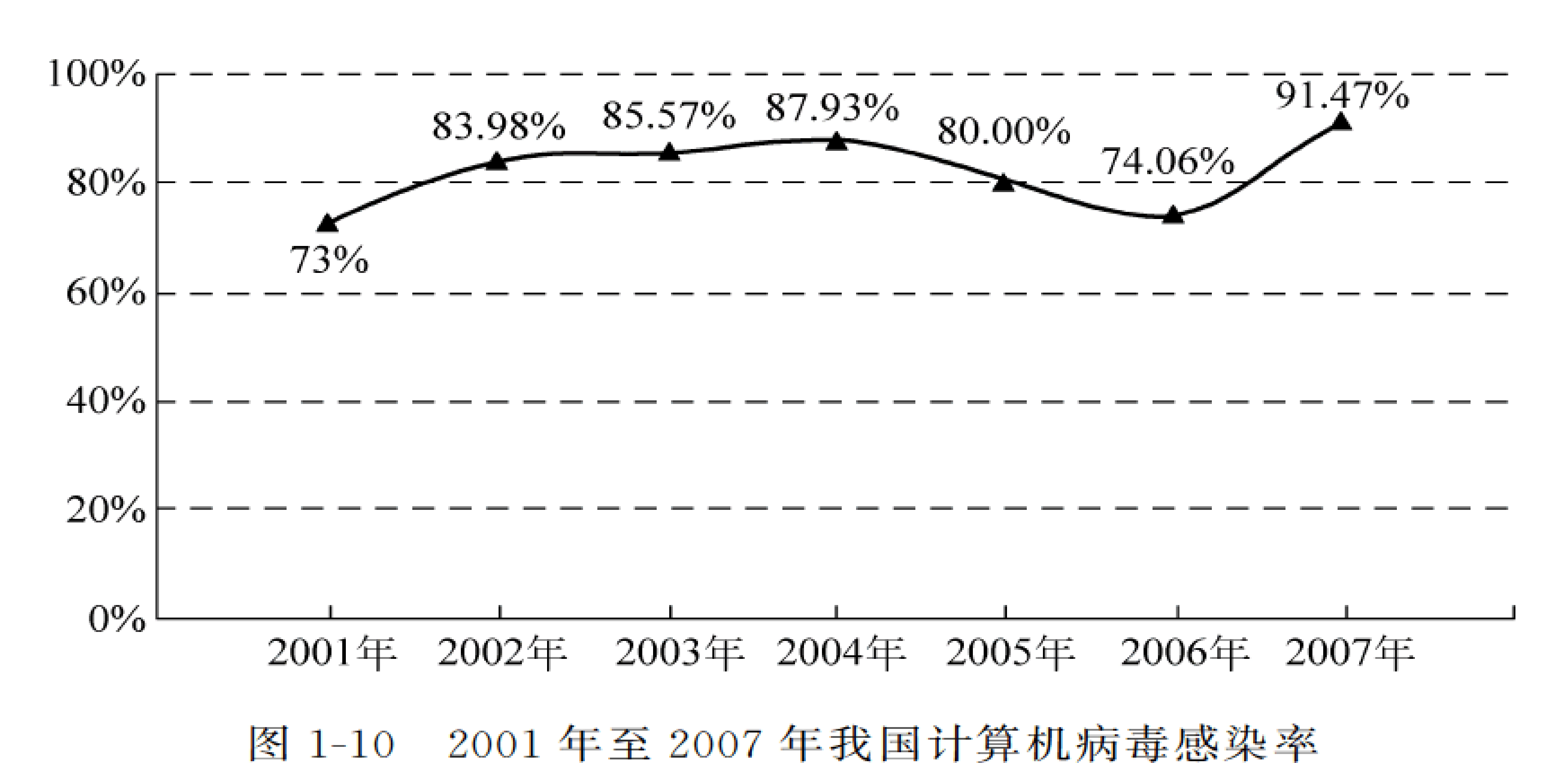



图 1-10　2001 年至 2007 年我国计算机病毒感染率

另据 CNCERT/CC 接收和监测的网络安全事件情况，2007 年我国内地被植入木马的主机 IP 数量增长惊人，是 2006 年的 22 倍。2007 年 CNCERT/CC 抽样监测发现感染木马控制程序的境内外主机数达 623 万个，其中我国内地有 362 万个 IP 地址的主机被植入控制程序，并有 1 万多个境外控制服务器对我国内地的主机进行控制。

我国网站的安全问题也十分严峻，大量网站被黑客入侵和篡改，甚至被植入木马攻击程序，成为黑客的得力工具。

从病毒造成破坏的情况来看，浏览器配置被修改、数据受损或丢失、系统使用受限、网络无法使用、密码被盗是病毒的主要破坏方式。近年来病毒功能越来越强大，不仅拥有蠕虫病毒传播速度和破坏能力，而且还具有木马的控制计算机和盗窃重要信息的功能。

同时，网上贩卖病毒、木马的活动不断增多，且公开化。利用病毒、木马技术传播垃圾邮件和进行网络攻击、破坏的事件呈上升趋势。因此，种种迹象表明，病毒的制造、传播者追求经济利益的目的越来越强，这种趋利性引发了大量的网络犯罪活动，威胁网络的应用与发展。

黑客的威胁见诸报端的已经屡见不鲜，内部工作人员的不小心甚至可能充当“间谍”。由于内部工作人员能较多地接触内部信息，工作中的任何不小心都可能给信息安全带来危险。这些都使信息安全问题越来越复杂。无论是有意攻击，还是无意的误操作，都将会给系统带来不可估量的损失。有些攻击者只是恶作剧地篡改主页面，或采用拒绝服务等攻击；另一些攻击者可以窃听网络上的信息，窃取用户的口令、数据库的信息，还可以篡

改数据库内容，伪造用户身份，否认自己的签名。更有甚者，攻击者可以删除数据库内容，摧毁网络节点，释放计算机病毒等。

综上所述，在享受网络提供的资源共享、信息交换与及时传递的好处的同时，网络必须有足够强的安全措施。无论是在局域网还是在广域网，无论是单位还是个人，网络的安全措施应是能全方位地针对各种不同的威胁和脆弱性，这样才能确保网络信息的保密性、完整性和可用性。

要想保持长久，就必须防患于未然。一方面，人类越来越依靠计算机，计算机被人们认为是 21 世纪每个人的必修课，人们的生活和工作已经日趋计算机化了；另一方面，计算机安全，尤其是网络安全也成了人们研究的课题。计算机犯罪作为一种更为隐蔽的犯罪手段给社会带来了很大危害，试想有人通过网络入侵到公司的一台计算机上，偷走了用户机密商业文件并卖给了竞争对手，后果不堪设想。当然，这只是一个简单的例子，计算机安全远远不止这个。

"网络就是计算机"，由此可见网络在计算机领域的重要地位。随着局域网技术不断走向成熟，网速在不断地提高，从 10Mb/s 以太网，100Mb/s 以太网，1000Mb/s 以太网，再到今天 10Gb/s 以太网。与性能不断提高相反，建网的价格却在不断下降。于是，大家纷纷组建自己的局域网，把自己的局域网连到了 Internet 上。

在软件方面，Unix 系统早在 20 世纪 80 年代就包含了 TCP/IP 系统，并获得了巨大的成功，Microsoft 公司把网络功能也嵌入了 Windows 操作系统，使 Windows NT 和 Windows XP，以及 Windows Vista 真正成为一种网络操作系统。

Internet 一方面给人们带来了经济上的实惠、通信上的便捷，但另一方面黑客和病毒的侵袭，又把人们置于了进退两难的境地。为了安全，有的干脆断开与 Internet 的连接。为了在保护自身网络同时又可以享受 Internet 带来的服务，人们设计了防火墙。它可以起到卫士的作用，守护着内部网络通向 Internet 的大门。

如今 Internet 的安全问题成了关注的焦点。对安全问题的忧虑，给认为 Internet 已经完全胜任商务活动的过高期望泼了一盆冷水，可能也延缓或阻碍了 Internet 作为国家信息基础设施或全球信息基础设施成为大众媒体。一些调查研究表明许多个人和企业之所以对加入 Internet 持观望态度，其主要原因就是出于安全的考虑。与此同时也有分析家警告商家，不加入 Internet 会有什么危害。尽管众说纷纭，但有一点几乎是大家都赞同的，那就是 Internet 需要更多更好的安全机制。

### 1.2.4　关于安全的权衡

安全，就是确保不发生有危害的事情。但实际上对于安全的含义，有非常复杂的解释，对这些解释进行剖析，有助于理解安全的真正含义。

通常情况下，某些传统和经验的方法比较适用于安全这个概念。首先，安全总是和效率有关。要想避免不安全的情况出现，最好就是什么事都不做。打个比方，存放在车库里的车不可能发生交通事故。但是，汽车不跑路并不是我们的目的。我们真正的目的是保障有利的事情发生，而避免有危害的事情。

其次，安全与危险是共存的。例如，前门加锁的有效性很大程度上取决于所要防卫的

窃贼类型。对于小贼,有一定的防卫意义,而对工具齐全、熟于此道的老贼则没有什么意义。类似地,在网络上防止普通的不安全事件发生,采用一般的安全技术就可以了,但是要防止蓄意的敌对方,就不是一件容易的事情。

第三,必须从整个系统的角度去考虑安全问题。系统的安全程度由系统的最薄弱环节决定。也就是说,只保证前门安全是不够的,狡猾的小偷会从所有可能疏于防备的地方潜入房子,尤其是远离装有牢固的锁的那些地方。类似地,在网络中的安全问题是一个全方位的问题,而不仅仅是某项单一的技术。

第四,实施安全的措施必须尽可能简单。试想,如果每次进门都需要花 30 分钟的艰辛去打开一个复杂的锁,可能谁也许不愿意锁门。正如专家所说,越复杂的系统就容易出错。使安全系统既简单又可靠是安全措施的宗旨。

第五,安全的实现必须注意性能价格比。例如,如果加锁的费用比所防护的内容的价值还昂贵,那将是毫无意义的。由于每个人对价值的认定是不同的,事情就变得很复杂。

# 1.3 网络中的协议

## 1.3.1 基本概念

在网络中,除了网络操作系统外,最重要的就是各种各样的网络协议。

网络能有序、安全运行的一个很重要原因就是它遵循一定的规范,也就是说,信息在网络中的传递同人在街上行走一样,也要用规则来约束和规范。网络里的这个规则就是通讯协议。换句话说,通讯协议是网络社会中信息在网络的计算机之间、网络设备之间及其相互之间“通行”的交通规则。

在不同类型的网络中,应用的网络通讯协议也是不一样的。虽然这些协议各不相同,各有优缺点,但是所有协议的基本功能或者目的都是一样的,即保证网络上信息能畅通无阻、准确无误地被传输到目的地。

通讯协议也规定信息交流的方式,信息在哪条通道间交流,什么时间交流,交流什么信息,信息怎样交流,这就是网络中通讯协议的几个基本内容。

在网络中,甲把文件 x 传送到乙的过程如下:

(1) 传送软件执行“命令”,向操作系统申请网络服务。

(2) 操作系统通过网络模块来为传送软件提供服务。

(3) 网络模块通过网卡的服务来为操作系统服务。

(4) 最终通过一系列由应用软件→操作系统→网络模块→网卡硬件的互相配合完成文件传送操作。

协议由语义、语法和时序三部分组成。语义规定通信双方彼此“讲什么”(含义),语法规定“如何讲”(格式),时序关系则规定了信息交流的次序(顺序)。

理论上,只要有一套协议即可,但由于网络技术在不断发展,应用领域在不断拓宽,加上历史的原因(20 世纪 70 年代各大计算机公司在网络领域“诸侯割据”,纷纷推出自己的

网络通信协议，既为网络技术的发展作出了贡献，也造成协议品种杂、多的局面），所以，目前尚无一套统一可用的网络协议。

正如理论上人类只要一种语言就可以相互沟通，但实际上却有许许多多的语言存在。

可以将人与人的"通信"分为 3 个相关的层次：认识层、语言层、传输层。请看下面的例子。

**例 1**　让一成都老妪与北京科学家进行如下的"通信"：

| | 成都老妪 | 北京科学家 | 结　　果 | 用网络术语表达结果 |
|---|---|---|---|---|
| 谈论内容 | 成都城内菜价 | 网络技术 | 不可理喻 | 认识层"协议"不兼容 |
| 所用语言 | 成都方言 | 英语 | 不知所云 | 语言层"协议"不兼容 |
| 通信方式 | 电话 | 计算机 | 无法联通 | 传输层"协议"不兼容 |

**例 2**　让一成都老妪与北京的成都籍科学家进行如下的"通信"：

| | 成都老妪 | 成都籍科学家 | 结　　果 | 用网络术语表达结果 |
|---|---|---|---|---|
| 谈论内容 | 成都城市变化 | 家乡情况 | √ | 认识层"协议"兼容 |
| 所用语言 | 成都方言 | 成都方言 | √ | 语言层"协议"兼容 |
| 通信方式 | 电话 | 电话 | √ | 传输层"协议"兼容 |

所以，人们为了能够彼此交流思想，首先需借助一种分层次的通信结构；其次，层次之间不是相互孤立的，而是密切相关的，上层的功能是建立在下层的基础上，下层为上层提供某些服务，而且每层还应有相应的协议规则。

网络通信情况同样如此，只是区分更细一些。

计算机网络理论把这整个过程定义成一个分层服务体系。在国际标准组织（ISO）的著名标准开放式系统互连参考模型（OSI）里，这个复杂的体系依次有应用层、表示层、会话层、传输层、网络层、数据链路层、物理层，称为 ISO 七层模型，参见表 1-1。

**表 1-1　ISO 七层模型**

| | |
|---|---|
| 应用层 | 网络的用户接口（如上网软件等） |
| 表示层 | 不同系统数据格式转换（如加解密） |
| 会话层 | 进程（执行中的程序）间会话管理与会话同步（"会话"即用户间连接） |
| 传输层 | 报文的正确传输（报文的生成、收发、组合与差错检查） |
| 网络层 | 路由选择和流量控制（选择 LAN 间传输路径） |
| 数据链路层 | 帧的正确传输（帧的生成、收发与差错检查） |
| 物理层 | 数据比特流（0、1）的正确传输（比特流的生成、收发与差错检查） |

加入分层的概念，是为了将整个体系的不同组成部分更好地按不同功能级别来划分；同时在层次中引入了服务、接口和协议这 3 个概念，服务说明某层为上一层提供什么功能，接口说明上层如何使用下一层的服务，而协议定义如何实现本层的服务。

实际使用中，协议并非严格按照这七层来定义，因为 OSI 七层参考模型是一个理论模型，实际应用则千变万化，因此更多把它作为分析、评判各种网络技术的依据；对大多数应用来说，只将它的协议族（即协议堆栈）与七层模型作大致的对应，看看实际用到的特

定协议是属于七层中的某个子层,还是包括了上下多层的功能,见表 1-2。

**表 1-2 TCP/IP 协议族与七层模型的对应关系**

<table>
<tr><th>OSI 七层模型</th><th>TCP/IP 协议族</th></tr>
<tr><td>应用层</td><td rowspan="3">应用层(HTTP、FTP、TELNET 等)</td></tr>
<tr><td>表示层</td></tr>
<tr><td>会话层</td></tr>
<tr><td>传输层</td><td>传输层(TCP/UDP)</td></tr>
<tr><td>网络层</td><td>网络层(IP)</td></tr>
<tr><td>数据链路层</td><td rowspan="2">网络接口层</td></tr>
<tr><td>物理层</td></tr>
</table>

TCP/IP 的多数应用协议将 OSI 应用层、表示层、会话层的功能合在一起,组成应用层,典型协议有 HTTP、FTP、TELNET 等;TCP/UDP 协议对应 OSI 的传输层,提供上层数据传输保障;IP 协议对应 OSI 的网络层;TCP/IP 的最底层功能由网络接口层实现,相当于 OSI 的数据链路层和物理层,TCP/IP 应用已有的底层网络实现传输,对该层并未作严格定义。

数据从上层往下层传送,往往要封装附加的信息(如目的地址、错误检测码等);数据从下层往上层传送,则需在对应层相应去除所封装附加的信息(还原)。

封装是指信息通过各层向下传递时,每层的软件负责加上它的报头(header)或报尾(trailer)信息(每层都要封装从高层来的信息),见图 1-11。

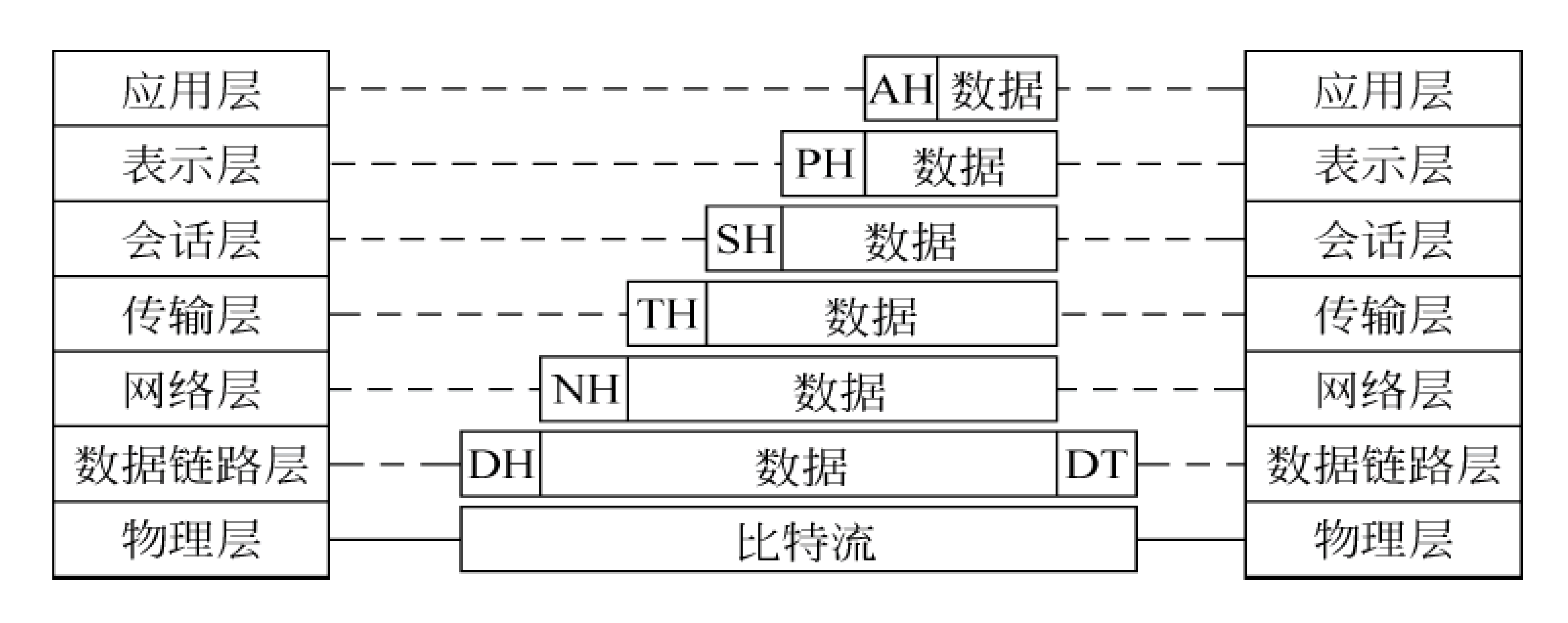



图 1-11 数据的封装

## 1.3.2 网络协议的定义

协议就是两个或者两个以上的参与者,为完成特定的任务而约定采取的一系列步骤。这个定义包含三层含义:

(1) 协议自始至终是有序的过程,每一个步骤必须执行,在前一步没有执行完之前,后面的步骤不可能执行。

(2) 协议至少需要两个(或两个以上)参与者。

(3) 通过协议必须能够完成某项任务。

综上所述,协议是一系列步骤,包括两方或多方,设计它的目的是要完成一项任务。"一系列步骤"意味着协议实现是从开始到结束的一个序列,每一步必须依次执行,在前一

步完成前，后面的步骤都不能执行；“包括两方或多方”意味着完成这个协议至少需要两个人，单独的一个人不能完成协议，当然单独的一个人也可采取一系列步骤去完成一个任务(例如烤蛋糕)，但这不是完成协议(另外一些人必须吃蛋糕才完成协议)；最后，“设计它的目的是要完成一项任务”意味着协议必须做一些事。有些东西看起来像协议，但不是完成一个任务，那也不是协议。

协议还有其他一些特点。

(1) 预先建立：协议执行中的每一方都必须了解协议，并且预先知道所要完成的所有步骤。

(2) 相互同意：协议中的每一方都必须同意遵循它。

(3) 非二义性：协议必须是不模糊的，每一步必须明确定义，并且不会引起误解。

(4) 完整性：协议必须是完整的，对每种可能的情况必须规定具体动作。

总之，协议是一系列的步骤，按照规定的步骤线性执行，除非指定它转到其他步骤。每一步至少要做下列两件事中的一件，即由一方或多方计算，或者是在各方中传送信息。

安全协议是使用安全技术(包括密码技术)的协议，协议的参与者可能是相互信任的人，也可能是完全不信任的人。安全协议使网络环境下相互不信任的通信参与方能够相互配合，通过安全连接和安全机制的实现来保证通信过程的安全性、可靠性和公平性。本书所指的安全协议特指使用密码学技术来完成某种安全功能的协议。

密码协议是使用密码学的协议。参与该协议的伙伴可能是朋友和完全信任的人，或者也可能是敌人和完全不信任的人。密码协议包含某种密码算法，但通常情况下，协议不仅仅是为了简单的秘密。参与协议的各方可能为了计算一个数值想共享它们的秘密部分、共同产生随机序列、确定相互的身份或者同时签署合同。在协议中，使用密码的目的是防止或发现偷听者和欺骗者，相互之间不信任的各方也能够在网络上完成这些协议。

后续几章将讨论许多安全协议。在其中的一些协议中，参与者中的一个有可能欺骗其他人，偷听者也可能暗中破坏协议或获悉秘密信息。一些协议之所以失败，是因为设计者对需求不是定义得很完备；其他一些失败是因为协议的设计者分析得不够充分。与密码算法类似，证明协议的不安全比证明协议安全更容易。

### 1.3.3　协议的目的

在日常生活中，几乎所有的事情都有非正式的协议：电话订货、玩扑克、选举投票。但是几乎没有人认真考虑过这些协议，这些协议随着时间的推移而发展，人们都知道怎样使用它们，而且也很有效。

越来越多的人通过网络交流，代替面对面的交流。计算机需要正式协议来完成人们不用考虑就能做的事情。如果从一个地方迁移到另一个地方，可能会发现玩扑克的方法与以前的方法有所不同，人们一般很容易就能适应它，但换成计算机就不那么灵活了。

许多面对面的协议依靠人的现场存在来保证公平和安全。举例来说，生活中有人会

交给陌生人一叠现金去替自己买食品吗？如果一个人没有看到另一个人洗牌和发牌，这个人愿意和那个人玩扑克吗？

那种假设使用网络的人都是诚实的想法是不切实际的。类似的不切实际的想法还有网络的管理员是诚实的，网络的设计者是诚实的等。

虽然绝大多数人是诚实的，但是不诚实的少数人可能招致很多危害。通过规定协议，可以查出不诚实的人企图欺骗的把戏，还可开发挫败这些欺骗者的协议。正如古人所说的："害人之心不可有，防人之心不可无。"

除了规定协议的行为外，协议还根据完成某一任务的机理，抽象出完成此任务的过程。例如，不管是 IBM PC 还是工作站机或者传真机，通信协议是相同的。我们考查协议，而不用局限于具体的实现上。当拥有一个好的协议时，从计算机到电话再到智能烘箱的所有事情都能够实现。

## 1.3.4 协议中的角色

在后续各章中，如果没有特殊说明，将采用表 1-3 所列的符号来帮助说明协议。A 和 B 是开始的两个人，他们将完成所有的两人协议。按规定，由 A 发起协议，B 进行响应。如果协议需要第三或第四人，C 和 D 将扮演这些角色。由其他人扮演的专门配角，将在后面介绍。

**表 1-3 协议参与者的表示**

| | | | |
|---|---|---|---|
| A | 所有协议中的第一个参加者 | M | 恶意的主动攻击者 |
| B | 所有协议中的第二个参加者 | T | 值得信赖的仲裁者 |
| C | 在三方协议中的参加者 | W | 监察人：在某些协议中保护 A 和 B |
| D | 在四方协议中的参加者 | P | 证明人 |
| E | 窃听者 | V | 验证人 |

## 1.3.5 协议的分类

一般地，根据协议的执行过程，可以将它分为 3 类：仲裁协议、裁决协议、自动执行协议，见图 1-12。

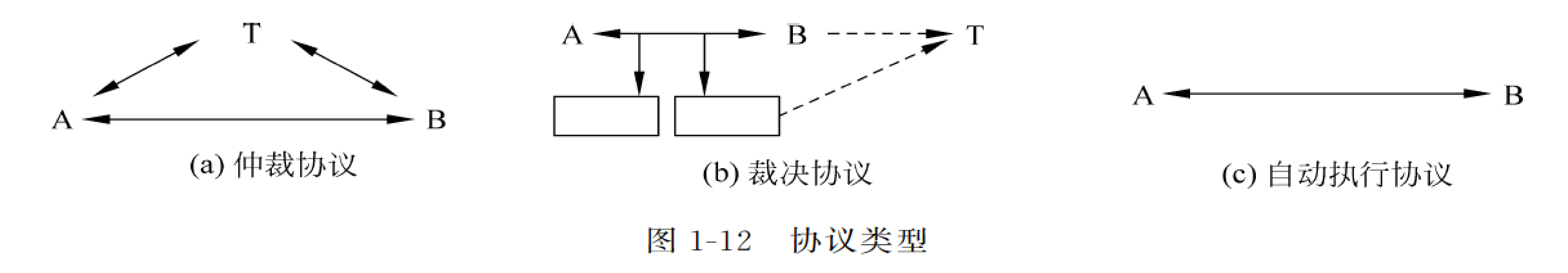



图 1-12 协议类型

在网络中，按照协议完成的安全功能，可以分为 3 类：密钥交换协议、认证协议、认证和密钥交换协议。这些安全协议将在第 3 章中讲述。本章的目的是建立安全协议的基本概念。

### 1. 仲裁协议

仲裁协议是在协议的执行过程中有仲裁者参与的协议，见图 1-12(a)。仲裁者是在完成协议的过程中，值得信任的、公正的第三方。

"公正"意味着仲裁者在协议中没有既得利益，对参与协议的任何人也没有特别的利害关系。"值得信任"表示协议中的所有人都接受这一事实，即仲裁者说的都是真实的，做的都是正确的，并且将完成协议中涉及自己的部分。仲裁者可以帮助互不信任的双方完成协议。

在现实社会中，律师经常作为仲裁者。举一个例子，A 要卖汽车给不认识的 B。B 想用支票付账，但 A 不知道支票的真假。在 A 将车子转给 B 前，她必须查清支票的真伪。同样，B 也并不相信 A，就像 A 不相信 B 一样，在没有获得所有权前，也不愿将支票交与 A。

这时就需要双方都信任的律师。在律师的帮助下，A 和 B 可以用下面的协议保证互不欺骗。

(1) A 将车的所有权交给律师。

(2) B 将支票交给 A。

(3) A 在银行兑现支票。

(4) 在等到支票认证无误能够兑现之后，律师将车的所有权交给 B。如果在规定的时间内支票不能兑现，A 将证据出示给律师，律师将车的所有权和钥匙交还给 A。

在这个协议中，A 相信律师不会将车的所有权交给 B，除非支票已经兑现；如果支票不能兑现，律师会把车的所有权交还给 A。而 B 相信律师有车的所有权，在支票兑现后，将会把车主权和钥匙交给他。而律师并不关心支票是否兑现，不管在什么情况下，他只做那些他应该做的事，因为不管在哪种情况下，他都有报酬。

在这个例子中，律师起着担保代理作用。

如果银行也使用仲裁协议，B 就可以用保付支票从 A 手中购买汽车：

(1) B 开一张支票并交到银行。

(2) 在验明 B 在银行存的钱足以支付支票上的数目后，银行将保付支票交与 B。

(3) A 将车的所有权交给 B，B 将保付支票交给 A。

(4) A 到银行兑现支票。

这个协议也是可行和有效的，因为 A 相信银行的证明。A 相信银行会将保存的 B 的钱给她，不会将她的钱用于其他业务。

公证人是另一种仲裁人。当 B 从 A 收到已公证的文件时，他相信 A 签署的文件是她(A)自己亲自签署的。因为如果有必要，公证人可出庭证实这个事实。

仲裁人的概念与人类社会一样悠久。总是有那么一些人——统治者、牧师等，他们有公平处理事情的权威。在现实社会中，仲裁者总是有一定社会地位和声望的人，而背叛公众的信任是很危险的事情。例如，视担保为儿戏的律师肯定会被开除出律师界。虽然现实世界里并不总是如此美好，但这种方式确实是理想的。

这种思想可以转化到计算机世界中，但计算机仲裁者有下面几个问题：

(1) 如果在知道对方是谁,并能见到对方的情况下,就很容易找到和相信中立的第三方。互相怀疑的双方很可能也会不信任在网络别的地方并不露面的仲裁者。

(2) 网络必须负担仲裁者的费用,就像我们知道的律师费用。谁来负担这种网络费用呢?

(3) 在任何仲裁协议中都有延迟的特性。

(4) 仲裁者必须处理每一笔交易。任何一个协议在大范围执行时,仲裁者是潜在的"瓶颈"。增加仲裁者的数目能缓解这个问题,但费用将会增加。

(5) 由于在网络中每个人都必须相信仲裁者,对试图破坏网络的人来说,仲裁者便是一个易受攻击的弱点。

尽管如此,仲裁者仍继续扮演其角色。在使用可信任的仲裁协议中,这个角色将由 T 来扮演。

### 2. 裁决协议

由于雇用仲裁者代价高昂,仲裁协议(参见图 1-12(b))可以分成两个低级的子协议:一个是非仲裁子协议,这个子协议是想要完成协议的各方每次都必须执行的;另一个是仲裁子协议,仅在例外的情况下才执行,即有争议的时候才执行,这种特殊的仲裁者叫做裁决人。

裁决人也是公正的和可信的第三方。裁决人不像仲裁者,并不直接参与每一个协议,只有需要确定协议是否被公平地执行时才将其请来。

法官是职业的裁决者。法官不像公证人,仅仅在有争议时才需要法官出场,A 和 B 可以在没有法官的情况下订立合同。除非他们中有一个人把另一人拖到法院,否则法官绝不会看到合同。

合同签字协议可以归纳为以下形式。

非仲裁子协议(每次都执行):

(1) A 和 B 谈判合同的条款。

(2) A 签署合同。

(3) B 签署合同。

裁决子协议(仅在有争议时执行):

(1) A 和 B 出现在法官面前。

(2) A 提出她的证据。

(3) B 也提出他的证据。

(4) 法官根据证据裁决。

裁决者和仲裁之间的不同是裁决者(在这本书中用的)并不总是必需的。如果有争议,法官被请来裁决;如果没有争议,就没有必要请法官。

已有计算机裁决协议。这些协议依赖于与协议有关的各方都是诚实的;如果有人怀疑出现欺骗时,一个中立的第三方能够根据存在的数据正文文本判断是否有人在欺骗。在好的裁决协议中,裁决者还能确定欺骗人的身份。裁决协议是为了发现欺骗,而不是为了阻止欺骗。发现欺骗是起了防止和阻碍欺骗的作用。

### 3. 自动执行协议

自动执行协议是协议中最完善的。协议本身就保证了公平性(参见图 1-12(c))。不需要仲裁者来完成协议,也不需要裁决者来解决争端。协议的构成本身不可能发生争端。如果协议中的一方试图欺骗,其他各方马上就能发觉并且停止执行协议。无论欺骗方想通过欺骗来得到什么,他都不能如愿以偿。

最好是让每个协议都能自动执行。但是,实际上没有一个协议是自动执行的。

## 1.4　网络协议面临的威胁

密码攻击可以直接攻击协议中所用的密码算法、用来实现该算法和协议的密码技术或者攻击协议本身。本节仅讨论协议,并假定密码算法和密码技术是安全的,只关注对协议本身的攻击。

可以采用各种方法对协议进行攻击。与协议无关的人偷听协议的一部分或全部,这叫做被动攻击。因为攻击者不可能影响协议,所有能做的就是观察协议并试图获取信息。这种攻击相当于密码学中的唯密文攻击。由于被动攻击难于发现,因此协议应阻止被动攻击而不是发现这种攻击。在这种协议中,偷听者的角色将由 E 扮演。

另一种攻击是改变协议以便对自己有利。攻击者可能假装是其他一些人,在协议中引入新的信息;或删掉原有的信息,用另外的信息代替原来的信息;或重放旧的信息;或破坏通信信道;或者改变存储在计算机中的信息等。这些叫做主动攻击,他们具有主动的干预行为。这种形式的攻击依赖于网络。

被动攻击试图获取协议中各方的信息。他们收集协议各方所传送的信息,并对它们进行分析。而主动攻击有更多的目的。攻击者可能对获取信息感兴趣,也可能希望降低系统性能,破坏已有的信息,或者获得非授权的资源存取。

相比之下,主动攻击更为严重,特别是在那些各方都不能信任的协议中。而攻击者不一定都是入侵者,可能是合法的系统用户,也可能是系统管理员。甚至有很多主动攻击者,他们在一起工作,每人都是合法的系统用户。这种恶意的主动攻击者的角色将由 M 扮演。

攻击者也可能是与协议有关的各方中的一方。他可能在协议期间撒谎,或者根本不遵守协议,这类攻击者叫做骗子。被动骗子遵守协议,但试图获取协议外的其他信息;主动骗子在协议的执行中试图通过欺骗来破坏协议。

如果与协议有关的各方中的大多数是主动骗子,就很难保持协议的安全性。但合法用户发觉是否有主动欺骗却是可能的。当然,协议对被动欺骗来说应该是安全的。

为了应对协议的威胁,必须设计安全的协议。

对于安全协议的定义:安全协议是建立在密码体制基础上的一种通信协议,计算机网络或分布式系统中的参与者通过安全协议,借助于密码算法来达到密钥分配、身份认证、信息保密以及安全地完成电子交易等目的。

安全协议是一种通信协议,它的主要目的是利用密码技术实现网络通信中的密钥分

配和身份认证。安全协议是网络通信安全系统的基础，是实现计算机网络安全的关键。然而，大量事实表明，有许多安全协议经过认真仔细地分析、设计和实现后仍然存在漏洞，有些甚至在使用许多年后才被发现有漏洞。

## 1.5 本章重点和难点

本章的重点是建立网络和协议的概念。

本章采用下述路线建立网络的概念：首先通过网络的构成和分类，以及网络的发展引入网络的基本概念，然后通过介绍网络安全的含义、不同应用环境中的网络安全问题、网络安全的重要性以及关于网络安全的权衡考虑因素，引入网络安全的概念。

在此基础上，通过下述路线建立网络协议的概念：通过介绍网络中的协议、网络协议的定义、协议的目的、协议中的角色以及协议的分类，介绍网络中的协议。

最后介绍网络协议面临的威胁。

本章的重点是了解掌握网络协议的基本概念。

## 习题与思考题

1. 试述网络的基本构成及相互之间的联系。
2. 网络在不同标准下的分类有哪些？
3. 论述网络的发展。
4. 论述并分析造成网络不安全的主要原因。
5. 论述网络安全的基本内容。
6. 试述网络安全的重要性。
7. 怎样对安全进行权衡？
8. 试述密码学与协议攻击之间的联系。
9. 试述网络协议的基本内容。
10. 网络协议的目的是什么？
11. 协议的基本类型有哪些？
12. 描述仲裁协议、裁决协议、自动执行协议三者之间的联系与区别。
13. 有哪些协议攻击方式？协议的安全性依赖于哪些因素？

# 第 2 章　Internet 的协议

本章的目的是通过介绍现有的网络及其协议，使大家对网络协议及其安全性有一个感性的认识。

本章有 5 个小节。第 2.1 节首先对 TCP/IP 协议进行概括的介绍，供那些尚不熟悉 TCP/IP 概念的读者参考；第 2.2 节分析导致 Internet 不安全的原因；第 2.3 节介绍在 Internet 中与安全相关的协议，讨论在网络各层实施安全措施的优点与缺点；第 2.4 节介绍网络层的安全协议；第 2.5 节是本章重点和难点分析。

## 2.1　Internet 协议的基本构架

Internet 的基石是 TCP/IP 协议，该协议在实现上力求简单高效。

在 20 世纪 80 年代，人们普遍使用的还有另外一些网络协议体系——ISO（国际标准化组织）的 OSI、IBM 公司的 SNA 以及 DEC 公司的 DECnet 等。然而，所有这些协议没有一个是简单的，也不像 TCP/IP 那样是开放的。正是由于这个原因，TCP/IP 协议才得到了广泛的实施、开发和支持。

所有网络协议体系均包括下述基本组件。

（1）协议堆栈：由相互间通信、高效率传输数据包的各层构成。

（2）定址系统：提供独一无二标识一个目的地（目标主机）的能力。为了实现大范围内的通信，有必要将通信实体唯一地标识出来。

（3）路由：决定一个特定数据包的传送路径，令其最终抵达目的地，这就是所谓的“路由选择”（routing）。

### 2.1.1　协议堆栈

TCP/IP 协议堆栈由如图 2-1 所示的 4 个层构成。堆栈中的每个层都有明确定义的功能及用途。每一层都能导出经准确定义的接口，在它上面或下面的层可通过该接口与之通信。这种分层结构具有多方面的优点。除了能简化协议堆栈的设计之外，还能简化它的使用。

设计之所以能得到简化，是由于每个层都只与紧靠在它上面或下面的层打交道。只要一个层提供的服务确定下来，相应的接口也会确定下来，所以每个层都可以独立设计。之所以说它的使用得到简化，是由于对最终使用网络堆栈的应用来说，完全可以不必顾及高度复杂的堆栈联网

| |
|---|
| 应用层 |
| 传送层 |
| 网络层 |
| 数据链路层 |

图 2-1　TCP/IP 协议堆栈分层

问题。换言之,复杂的技术性问题对网络应用来说是“透明”的。

## 2.1.2 数据流分析

从发送源到接收目的地,数据流如图 2-2 所示。

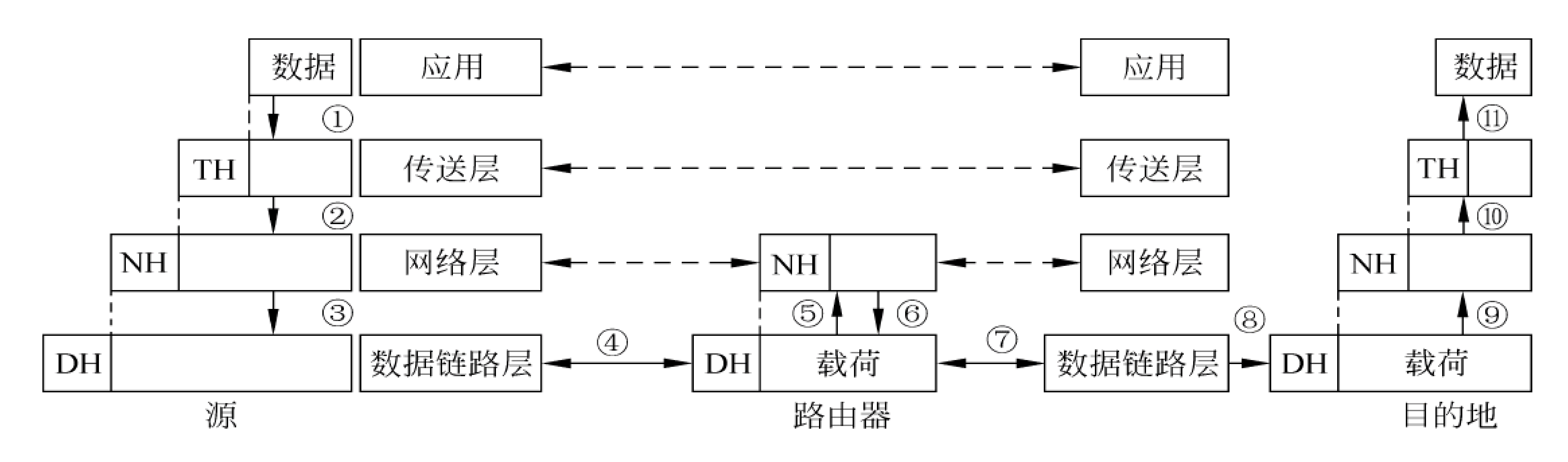



图 2-2 数据流

为讨论方便起见,首先假定传送协议是 TCP,而网络协议是 IP。基于这个前提:

(1) 源主机(起始主机)上的一个应用(程序)通过套接字接口将需要传给目的地的数据传给传送层。应用标定出希望与之通信的那个目的地。在这个“目的地”中,同时包括了目标主机以及在上面运行的应用(程序)。

(2) 传送层(这里是 TCP)取得这些要发送出去的数据,将一个传送头(这里是 TCP 头)追加到载荷,然后将整个载荷向下传给网络层。TCP 头里的字段有助于提供应用程序需要的服务。

(3) 网络层收到来自传送层的载荷。这个“载荷”(payload)内同时包含了数据以及 TCP 头。它进一步在载荷外追加一个 IP 头。然后,将载荷连同 IP 头向下传递给数据链路层。

(4) 数据链路层在来自网络层的载荷外追加一个数据链路头。数据链路层标识出下一站的物理地址,最后发出这个数据包。

(5) 下一站(假设下一个是“转发”主机而不是目的地)的数据链路层收到数据包,剥下数据链路层头,将剩余的数据包向上传递给网络层。

(6) 网络层检查网络头,决定这个包需要传递的下一站,以便最终路由到目的地,再将其打回数据链路层。

(7) 数据链路层将数据链路头追加到载荷,将这个包传给下一站(下一个“转发”主机)。

(8) 步骤(6)和(7)不断重复,直到这个包最终抵达目的地。

(9) 抵达目的地之后,数据链路层仍然会剥下数据链路层头,再将其向上传给网络层。

(10) 网络层从包中剥下网络头,再将其向上传给传送层。

(11) 传送层检查传送头,核实已正确地为应用提供了服务。然后剥下传送头,检查这个包的目标是哪一个应用,再向上传递给应用。

(12) 位于目的地的应用(程序)最终收到由源应用传给自己的数据。

## 2.1.3　网络层和传送层

在 TCP/IP 协议族中，共存在两种网络协议：IPv4 和 IPv6。下面将在 2.1.4 节中讨论。

在传送层，TCP/IP 协议族实现了两种协议：传输控制协议（TCP）以及用户数据报协议（UDP）。

TCP/IP 是一种“面向连接”的协议，可确保数据包的发送顺序以及保证它们的正确投递。其内建的机制可向应用层提供这些服务。除此以外，TCP 也实现了像“流控制”这样的机制，可确保目标主机不至于受到大量数据包的“狂轰滥炸”。UDP 则是一种“无连接”的协议，既不能担保数据包投递的顺序，也不支持流控制。至于到底选用 TCP 还是 UDP，则完全由应用（程序）决定。

尽管在此不讨论 TCP 和 UDP 头的细节，但还是要强调一下两种头都包含有的两个字段：“源端口”和“目标端口”字段。根据这两个字段提供的信息，便可知道目标主机在接收到数据时，如何对其进行处理。

TCP/IP 协议族指出一个数据包要传给哪个目标应用（程序），具体的信息用由连续五个单元构成的一个字元组表示：＜源地址，目标地址，源端口，目标端口，协议＞。对主机上运行的每个应用（程序）来说，这个字元组都必须是独一无二的。

现在讨论源和目标地址字段。这些字段是在网络头里设置的。源和目标端口均是 16 位的一个字段，在传送头内设置。源端口由源主机分配，而目标端口由目标主机分配。一个应用要想同另一个主机上的另一个应用通信，必须知道三件事情：目标主机的地址，应用程序正在上面运行的那个端口号以及通过哪种协议与之通信。举个例子来说，大多数 Web 服务器都在监听端口 80，并使用 TCP 协议。一个应用（程序）要同源和目标端口绑定到一起，而且还要指定通信时使用哪一种传送协议。传送协议利用这个字元组来标定由哪个应用（程序）负责数据的接收。

## 2.1.4　定址

对任何一种网络协议来说，“定址”（addressing）都是其至关重要的一项任务。下面讨论两种协议的定址方法。

### 2.1.4.1　IPv4

IPv4（互联网协议第 4 版）是当今最流行的一种网络层协议。它采用一种简单的定址方案，并提供了“无连接”服务。IPv4 是一种颇为成熟的路由框架。

IPv4 对每个主机都用一个 32 位的地址来标识。该地址通常以 A.B.C.D 的形式标识。通常将这种记号方法称为“点-十（进制）记号法”，因为每个符号（A，B，C 和 D）都是与一个字节（8 位）对应的十进制数字。IPv4 地址的一个例子是 128.127.126.125。之所以要选用这种表示方法，而不是用一个普通的大数字，是由于根据它便可知道地址的分级，而且易于理解。

IP 地址由两部分构成：一个网络 ID（标识符）以及一个主机 ID。其中，网络 ID 从逻

辑意义上将一系列 IP 地址组合到一起。之所以要进行这样的分组,是为了便于提供有效地路由选择以及其他服务,比如 IP 广播等。IP 地址的网络 ID 部分是通过 IP 地址与网络掩码的“逻辑与”(AND)运算获得的。网络掩码总是一系列相邻的二进制位 1。比如 255.255.255.0,255.255.0.0 以及 255.254.0.0 等,它们都是“网络掩码”(network mask)的典型例子。在这些例子中,从左数(自有效位开始)的 24、16 和 15 位均分别由 1 构成。比如对 128.127.126.125 这个 IP 地址来说,以上面的三个网络掩码为例,经过逻辑与运算后,它的网络 ID 部分分别为 128.127.126,128.127 以及 128.126。大家不妨计算一下,将不同的网络掩码与这个固定的 IP 地址进行逻辑与运算,结果是否和上面一样。IP 地址通常要和它的网络掩码联合起来表达,表达方式有两种:一种是 128.127.126.125/255.255.255.0,另一种是 128.127.126.125/24。两种表达方式意思完全一样,只是后一种非常简单——24 意味着网络 ID 是 IP 地址中 24 个有效位(即网络掩码从左数有 24 个 1,即 255.255.255.0)。

IPv4 头的结构如图 2-3 所示。

| 0 | 8 | | 16 | 1931 |
|---|---|---|---|---|
| 版本 | 头长度 | 服务类型 | 总长 | |
| 标识 | | | 标志 | 分段偏移 |
| 存活时间 | | 协议 | 头校验和 | |
| 源地址 | | | | |
| 目的地址 | | | | |
| IP 选项(如果有) | | | | |

图 2-3 IPv4 头的结构图

其中,经常用到的字段的用法如下所示。

“版本”(version)字段:这个 4 位长的字段用于指出版本号。对 IPv4 来说,它的值为 4。版本字段通常作为向后兼容的一种保证来使用。若定义了新版本,则需根据它同原来的系统进行沟通。

“头长度”(len)字段:此字段 4 位长,以 32 位(4 字节)为单位,指出头的长度。这样便把 IPv4 头的最大长度限制在 64 字节。事实上,这正是在开发一种新版本的 IP 时,需要解决的一项重要限制。

“服务类型”( type of service,TOS)字段:TOS 用于指出数据包的通信要求,8 位长。该字段的具体标准目前正在 IETF 的审查之中。

“总长”(total length)字段:按照网络字节顺序,16 位长,以字节为单位,指出数据报的长度(包括头在内)。该字段用于向接收端的网络层报告一个数据报的总长。

“标识”(identification)字段:这是一个 16 位的字段,用于唯一地标识出一个 IP 数据报。“IP 数据报”是指在传送载荷的基础上,加上 IP 头,在端主机的背景下使用。大多数情况下,标识字段都用在后文要讲述的“分片”的场合下。使用标识字段,可将哪个 IP 包属于一个 IP 数据报唯一地标识出来。

“标志”(flag)字段:在旗标占用的 3 位中,仅有 2 位有定义。第一位用来指出是否对 IP 包进行“分片”(fragment)。若将其设置为 1,路由器就会向主机返回一条控制消息,指

出它的 MTU(最大传输单元)是多大。这个位在 PathMTU(路径 MTU)中使用。对末端主机来说,它会通过一个专门的进程,计算出自己应生成多大长度的 IP 包,使数据包不至于要先进行分片,才能路由到目的地。这样做是必要的,因为对整个网络的运作来说,过多的分段会对其效率造成严重的损害。假如一个 IP 包分段在中途丢失了,那么传送层必须重新传送整个数据报。第二位则用来指出数据包是一个分段数据报的最后一个分段,还是紧接着有更多的分段。该位在分段数据包的重新组合、装配过程中起着重要的作用。

“分段偏移”(fragment offset)字段:该字段指出在一个 IP 数据报中,IP 包的偏移量是多少。该字段的实际应用将在“分段”小节详述。

“存活时间”(TTL)字段:该字段用于避免数据包在网络中无休止地游荡,并能从管理的角度出发,规定一个包的传输范围。主机会将该字段设为一个特定的默认值。在其以后的传输过程中,经过的每个路由器都会将该值减 1。假如路由器发现一个数据包的 TTL 值已变成 0,便会毫不留情地将其丢弃。这对避免重复路由颇有好处,因为假如没人把它丢掉,它就会在网络里无休止地游荡下去。

“协议”(protocol)字段:这个 8 位的字段用来指出该 IP 包所负载的传送协议。这个字段由端主机使用,在不同的传送协议中分解出真实的数据包。

“头校验和”(header checksum)字段:指出 IP 头的校验和是多少,用来保证 IP 头经过网络传输后的完整性。但这个校验和并不是一种经过加密的校验和,所以也很容易被仿冒。

“源地址”(source address)字段:这是一个 32 位的字段,指出生成这个包的源主机的 IP 地址。

“目的地址”(destination address)字段:同样是一个 32 位的字段,指出目标主机的 IP 地址。

“IP 选项”(IP option):这是一个可选字段,用来负载一些额外的信息。由于仅从 IP 安全的角度出发,这个选项并不重要,所以这里不作讨论。

#### 2.1.4.2 IPv6

IPv4 将地址空间限制在了 32 位以内。为避免地址资源全面用完的尴尬局面,IETF 早就开始了下一版 IP 协议的研究,名为 IPv6。IPv6 的一个主要优点是将地址空间从 32 位提升到了 128 位(16 个字节)。

在 IPv4 基础上作出的其他改动包括:

(1) 不再为通过路由器的数据包提供分段支持。现在要由端主机自己来执行 PMTU。

(2) 支持更丰富的选项。各选项现在以独立的头的形式存在,而不是成为 IP 头的一部分,显得更加灵活。我们称为“头链接”。在这种情况下,IP 扩展头(选项)要插入 IP 头与传送层头之间。

从概念上说,IPv6 无论地址还是子网的定义,本质上都和 IPv4 差不多。但是,IPv6 的地址长度达到了 128 位,而且采用了不同的表达方法,不再采用 IPv4 的点-十记号方

法，IPv6 采用的是一种完全不同的记号法，即由冒号分隔的十六进制数字，就像下面这样：

0123:4567:89ab:cdef:0123:4567:89ab:cdef

当然，可以有许多不同的变化，对这种地址加以压缩。建议读者参考有关 IPv6 定址结构的 IETF 文件，了解更详细的情况。

在 IPv6 中，网络掩码以及子网的概念都与 IPv4 类似。IPv6 运用了一种包容力更强的分级结构来进行定址，从而简化了路由选择以及地址分配的问题。

IPv6 头的结构如图 2-4 所示。

| 0 | 4 | | | 31 |
|---|---|---|---|---|
| 版本 | 通信类别 | 流标签 | | |
| 载荷长度 | | 下一个头 | 跳跃限制 | |
| 源地址 | | | | |
| 目的地址 | | | | |

图 2-4 IPv6 头的结构图

对各字段的解释如下。

“版本”(version)字段：这个长度为 4 位的字段用于指出版本号。对 IPv6 来说，它的值为 6。

“通信类别”(traffic class)字段：这个 8 位的字段用来指出数据包的通信要求，类似于 IPv4 头中的 TOS(服务类型)字段。

“流标签”(flow label)字段：这是一个 20 位的字段，仍处在试验阶段，目前仍不清楚它未来将用于何种目的。一种可能的用法是标识出需要由路由器特殊处理的一个流。

“载荷长度”(payload len)字段：这个 16 位的字段指出除 IPv6 头以外的载荷长度。

“下一个头”(next header)字段：这是一个长度为 8 位的字段，类似于 IPv4 的“协议”字段，唯一的差别是，和 IPv4 不一样，选项头可能会在这个字段中出现。

“跳跃限制”(hop limit)字段：这个 8 位的字段等同于 IPv4 头中的 TTL(存活时间)字段。

“源地址”(source address)字段：这是一个 128 位的字段，代表 IPv6 源地址。

“目的地址”(destination address)字段：同源地址字段，用 128 来表示 IPv6 目标地址。

#### 2.1.4.3 域名系统

域名系统(domain name system，DNS)是 Internet 中一种简单和易于扩展的目录系统。它最重要的功用是将采用 www.xyz.com 这种形式的一个机器名称翻译成数字形式的 IP 地址。对普通人来说，显然不可能记住大量地址，特别是当地址没有语义上的关联时。此外，IP 地址大多是动态分配的，每次都有可能不同。因此，IP 地址的记忆通常是件没有意义的事情。但在另一方面，网络层能够识别和处理的只能是这种形式的地址。因此，有必要设置一个易于伸缩的系统，将一个名字动态地翻译成 IP 地址。

为满足这个要求，DNS 定义了一个域名结构。

使用了“根”的概念。根服务器是众所周知的。事实上，目前有几个著名的根服务器。1 个为主根服务器，放置在美国。其余 12 个均为辅根服务器，其中 9 个放置在美国；欧洲 2 个，位于英国和瑞典；亚洲 1 个，位于日本。所有根服务器均由美国政府授权的互联网域名与号码分配机构 ICANN 统一管理，负责全球互联网域名根服务器、域名体系和 IP 地址等的管理。在“根”的下方，便是一些大家熟悉的“域”，如图 2-5 所示。

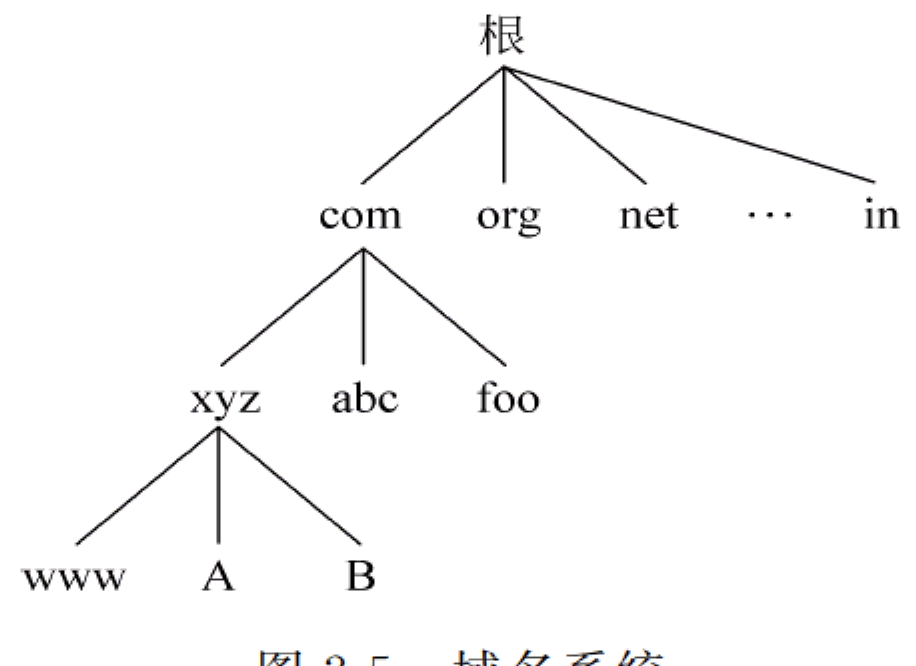


图 2-5 域名系统

在这些大家熟知的域下方，存在着多组织，均从属于特定的域。例如，com 域注册在根服务器下方，而公司 xyz 注册在 com 域下方。域名注册便意味着一个单位打算运行一个域名服务器，用来回应查询。

假如一个客户机想同主机通信，便会请求解析器（负责将域名解析成 IP 地址的软件）将对方的域名解析成 IP 地址。例如，在浏览器地址栏中输入 www. xyz. com，浏览器便会向解析器发出请求，将 www. xyz. com 这个名字“映射”成为 IP 地址。解析器随之会采取一系列预先规定的步骤，将域名解析成 IP 地址。解析好域名后，客户机便能用数字形式的 IP 地址同 www. xyz. com 处的 Web 服务器正常通信。而在操作者的眼中，似乎仍在使用普通的英语形式的地址。也就是说，域名解析在操作者的眼中是“透明”的。

除解析 DNS 名字以外，DNS 还用来解析邮件主机。换言之，它能判断将投递给 xyz . com 这家公司的所有邮件具体投递至何处。另外，DNS 也在不断地进行改进，可提供其他更高级的目录服务。在此建议读者参考其他书籍，以便更好地理解 DNS 以及它的功用。

## 2.1.5 路由

### 2.1.5.1 分段

所谓“分段”(fragmentation)是指在源主机或者路由器处，将一个 IP 包分割为多个 IP 包的一种过程。经过分段后，每个包都单独传送，并在目的地（目标主机）处重新装配。现在，大家马上会产生一个问题：为什么需要对 IP 包分段呢？

IP 层可接收来自传送层的任何载荷。传送层载荷可以为任意大小（仅受系统的缓冲区大小设置所限）。IP 层没有能力对上层传给自己的数据进行限制。然而，由于网络层(IP)必须通过一个物理接口来发送数据包，所以就对能够通过这个接口发送的数据包的大小提出了限制。这是由于对数据包通过的传输介质来说，它本身便存在物理上的一些限制。比如对 Ethernet（以太网）来说，包的最大长度不可超过 1518 字节。假如 IP 层的传送载荷大于 1480 字节（没有额外的选项设置），那么在它发送之前，必须先进行一次分段处理。否则，整个包的大小就会超过 1518 字节：(1480＋20 字节的 IP 头)＋(14 个字节的 Ethernet 层头)＋(4 字节的 CRC 错误校验)。

网络层数据包也有可能在路由器处进行分段处理，因为两个物理层之间可能出现数据包允许的最大长度不相符的情况。举个例子来说，路由器的数据进入接口连接的可能是一个令牌环网络，其最大包长度为 4096 字节；而外出接口连接的是一个以太网，其最大包长度为 1518。假如自令牌环进入的 IP 包超过 1500 字节，路由器就必须对其进行分段，尽管它自己本身并不产生新的 IP 包。

到达目的地以后，IP 层必须重新装配（整合）所有分段的数据包。完成之后，才能将载荷传给传送层。如图 2-2 所示，位于目标主机的传送层应看到和源主机自传送层向下传给网络层的一模一样的数据。因此，现在要由 IP 层来负责装配所有分段的数据包、构建传送载荷以及将其上传给传送层。

IP 层为达到分段及重新装配之目的，需要用到 IP 头内包含的旗标、长度以及分段偏移等字段。

但我们认为数据包的分段对于整个网络来说是有害的。这是由于，即便只丢失了其中一个分段的数据包，可靠传送层也必须重传整个传送载荷，因为 IP 层并不会对数据包进行缓存处理。为避免分段现象的发生，IP 层必须在从源主机到目标主机的整个路径中，监视 MTU（最大传输单元）值的大小。这个过程叫做 PathMTU 查找或者 PMTU。IP 层完成了对 PMTU 的调查以后，便永远不会发出超过 PMTU 上限的一个数据包，从而达到了避免分段的目的。

#### 2.1.5.2 ICMP

ICMP 是“互联网控制消息协议”的简称，用于确保正确的网络运作方式以及调试。该协议运行于网络协议（如 IPv4 或 IPv6）的顶部。

ICMP 消息同时可由主机及路由器产生，用来监视网络以及保证网络的正常运转。例如，假定一个路由器没有到达一个特定网络的路由，便会向主机返回一条 ICMP 消息，告知它网络不可抵达。假如路由器在丢弃一个数据包的时候，不作出任何明确的提示，那么对网络的监视无疑会成为一场噩梦。ICMP 的作用便是判断一个主机是否能够抵达。ICMP 在 PMTU 查找过程中得到了广泛应用。假如路由器需要对一个包进行分段，但包内未设置“分段”字段，路由器就会向始发主机（源主机）返回一条 ICMP 消息，告诉它这条链路的 MTU 值是多少。收到这条消息后，主机就知道如何生成数据包，保证它的大小不会超过规定的 MTU 值。

#### 2.1.5.3 多播

IP 也提供了将一个包发给 Internet 任何地方的多个主机的能力。这属于广播通信的一种特殊情况。在这种情况下，只有感兴趣的主机才会接收到这个包。以每次收视付费的视频广播为例，假定一家有线电视公司通过 Internet 将节目广播给自己的所有付费收视户，那么它可有 3 种选择来进行操作：

（1）将视频信号单独传送给每个收视户。这种方式会产生非常高的开销，因为相同的数据需要不断地复制，传送给每一位收视户。这样便会带来许多副作用，除了会显著增加 Internet 通信量以外，也会为负责传送数据的服务器带来可怕的压力。

（2）将信号广播到整个 Internet。这种方式是最不能让人接受的，因为即使那些没有付费的人，也会看到这个收视付费的节目。另外，Internet 一级的广播通信几乎是无法实现的。

（3）使用一种名为“多播”（多点传送）的技术，只向收视户传送视频信号。多播技术实际上是一种智能的包投递机制，只有真正的注册收视户才能得到数据包。数据只在那些含有收视户的信道上传送，只有那些真正注册的用户（节点）才能收到数据，而且同样的数据不必重复发送。

多播包的格式与单播 IP 包类似。但是，IP 头的“目的地”字段填的却是一个多播地址，而非单播地址。这样便产生了一个显而易见的问题：如果不能将一个主机唯一地标识出来，那么如何决定将数据包投递到哪里呢？这正是多播技术的妙处！路由层为此提供了特别的支持，知道哪些主机正在监听一个特定的多播地址。然而，对于多播（多点传送）技术的详细讨论已超出了本书的范围，这里不再介绍。

## 2.2　导致 Internet 不安全的原因

Internet 的基石是 TCP/IP 协议，该协议在实现上力求简单高效，没有考虑安全因素。所以，TCP/IP 本身在设计上就是不安全的。

现存 TCP/IP 协议的安全缺陷主要表现如下。

（1）容易被窃听和欺骗：TCP/IP 协议数据流采用明文传输，特别是在使用 FTP、TELNET 和 HTTP 时，用户的账号、口令都是明文传送的，很容易被在线窃听、修改和伪造。可以实现这些的工具很多，而且在网上是免费提供的。

（2）脆弱的 TCP/IP 服务：很多基于 TCP/IP 的应用服务都在不同程度上存在着安全问题。这很容易被一些对 TCP/IP 十分了解的人所利用，一些新的处于测试阶级的服务有更多的安全缺陷。这类例子如下。

① 数据源地址欺骗：攻击者修改发送的数据源地址，冒充某个可信节点的 IP 地址进行攻击。

② 路由信息协议攻击：路由信息协议用于在网络中发布路由信息，为网络中的节点提供一致的路由选择和可达信息，但节点对收到的信息不进行真假检验，因此，攻击者可以在网上发布错误的路由信息，利用 ICMP 的重定向信息欺骗路由器或主机，从而对网络进行攻击。

③ TCP 序列号欺骗：由于 TCP 序列号可以预测，因此，攻击者可以构造 IP 包对网络的某个可信节点进行攻击，通过对节点发送大量 IP 包，使节点消耗大量资源，致使系统崩溃，造成拒绝服务攻击。

（3）缺乏安全策略：许多站点在防火墙配置上无意识地扩大了访问权限，忽视了这些权限可能会被内部人员滥用。黑客从一些服务中可以获得有用的信息，而网络维护人员却不知道应该禁止这种服务。这类例子如下。

鉴别攻击：目前，防火墙系统能对 IP 地址、协议端口进行鉴别，而无法对登录用户身份的有效性进行鉴别，因此，不能对内部网络的入侵者进行有效防范。

（4）配置的复杂性：访问控制的配置一般十分复杂，所以很容易被错误配置，从而给黑客以可乘之机。例如，源路由选择欺骗，为了测试，IP 数据报设置了一个 IPSourceRouting 选项，修改该选项可以直接指明到达节点，攻击者可以使用这一选项进行攻击。

除上面的 4 个问题外，还有 TCP/IP 协议是公开的，了解它的人越多，那么就越有可能被人破坏。

现在，银行之间在专用网上传输数据所用的协议都是保密的，这样就可以有效地阻止入侵。当然，人们不能把 TCP/IP 协议和其实现代码保密，这样也不利于 TCP/IP 网络的发展，但可以在其他方面弥补。

另一方面，还有来自于人的威胁和自然的破坏。在这两个因素中，人的破坏更为严重。自然的破坏可以通过数据备份和冗余设置来完成，但人的破坏却是防不胜防的。

自然灾害和事故包括硬件故障、电源故障、软件错误、火灾、水灾、风暴和工业事故等，它们共同的特点是具有突发性，人们很难防止它们的发生。减小损失的最好办法，就是备份和冗余设置。备份并不像看起来那样简单，对一些金融机构、大公司，数据备份量极大。冗余设置的应用很多，简单的可以在局域网上建两个域服务器复杂的双环结构网络，当任何一单环发生故障，它会自动形成一个单环继续工作。一些主干网的路由器也需要冗余备份，防止路由器发生事故。

人的破坏主要来自于黑客，网络犯罪现在已经成为犯罪学的一部分。这些罪犯知识水平高，危害性大而且隐蔽性很强。现在，在 Internet 上实行商业信息盗窃、银行抢劫越来越多，黑客不单是一些想显示自己计算机水平的好奇的大学生，更有一些专职的商业间谍。在美国，许多银行开始进入 Internet，用户可以在 Internet 上存取银行里的钱。纽约的道琼斯股票交易所也提供了 Internet 服务，股民可以在网上买卖股票。而这些服务对黑客来说确实是具有诱惑力的。除了对钱财的贪婪，有一些黑客的入侵是出于其他目的。

另一种危害很大的入侵，是不满被解雇的职员对内部网络的入侵。这种入侵危害大是因为入侵者对内部网络很了解。实施入侵的黑客的水平很不相同，最低层的入侵就是电子邮件炸弹，这种网络危害不算很大，只可能会造成拒绝服务；再深一层的入侵就是得到了一些不该有的权限，如偷看别人的邮件，也可能获得了有限的非法的写的权力。最高的一层是入侵者得到了 root 权限，对网络可以进行任意的破坏。有些高级的入侵者本身就是一些大机构的系统管理员或安全顾问。1994 年，美国一家大机构的安全顾问被逮捕了，他是一个很有影响力的系统管理员，也曾在 BELL 实验室工作过，但他却多次入侵了 Intel 公司的网络，最后被 Intel 公司的一个网络管理员发现了。

除了直接入侵外，各种病毒程序也是 Internet 上的巨大危险。这些病毒可以随下载的软件、Java 程序、ActiveX 控件进入公司的内部网络。虽然，现在有些防火墙声称具有防毒的功能，但新的病毒是旧病毒的变异品种，仍会溜进用户的网络。病毒程序中有一种被称为特洛伊木马的程序，这种程序表面上是无害的，具有很强的隐蔽性，但在背后破坏用户的网络。

人们对网络病毒的研究开始于 1988 年的 Morris 蠕虫事件。当时，这种疯狂自我复

制的病毒袭卷了 Internet，使人们突然意识到了网络病毒的存在。

现在，黑客已经不单兵作战了，他们有自己的组织、自己的站点，如果发现一个安全缺陷很快就会在黑客中间传开。所以说，Internet 上的网络安全对抗并不是一次战斗，而是一场战争。

# 2.3　Internet 中与安全相关的协议

## 2.3.1　实施安全保护的层次

在 Internet 中，存在着大量特制的协议，专门用来保障网络各个层次的安全。决定到底在堆栈的什么地方应用安全措施时，要依赖于应用对安全保密的要求以及用户自己的一些需要。但无论在堆栈的什么地方采取安全措施，下面这些基本服务都是必须要提供的：

（1）密钥管理。

（2）机密性。

（3）不可抵赖。

（4）完整性/身份验证。

（5）授权。

网络的安全并非只是 OSI 参考模型某一层上的事情，事实上，每一层都可以采取一定的措施来防止某些类型的网络入侵事件，在一定的程度上保障数据的安全。比如在物理层上，包括在电缆的密封套中充入高压的氖气，当入侵者刺破密封套进行线路窃听时，由于气体泄漏导致压力下降而发出警告；在链路层上可以在两端进行加密解密；在网络层上可以采取防火墙技术过滤数据包；而在传输层上整个连接都可以被加密。但所有这些措施都只是针对数据加密。

从技术上看，在堆栈的什么地方实施安全措施，可进行选择。某些情况下有必要在某一层提供部分安全服务，而在另一层提供其他服务。

下面讨论在堆栈各层（应用层、传送层、网络层、数据链路层）提供安全保障的优点与缺点。

## 2.3.2　应用层

应用级的安全措施必须在端主机上实施。在应用层提供安全保障有以下几个方面的优点：

（1）由于是以用户为背景执行，所以更容易访问用户凭据，比如私人密钥等。

（2）对用户想保护的数据具有完整的访问权。这便简化了提供一些特殊的服务，比如不可抵赖。

（3）一个应用可自由扩展，不必依赖操作系统来提供这些服务。通常，应用对于操作系统上实施的东西没有控制权。

(4) 应用对数据有着充分的理解,可据此采取相应的安全措施。

应用层安全的缺点在于针对每个应用,都要单独设计一套安全机制。这意味着对现有的应用来说,必须对其进行改进,才能提供安全保障。由于每个应用都必须定义自己的安全机制,所以犯错误的概率大增,为黑客打开了更多的安全漏洞。

在应用中实施安全机制时,程序要和一个特殊的系统集成到一起,建立最终的安全机制。此类系统的例子包括 PGP、Kerberos 以及 SecureShell(安全外壳)。这些系统均属应用级的协议,可提供密钥协商以及其他安全服务。应用程序通过改进,可调用这种系统以使用它们的安全机制。典型的例子是 E-mail 客户端软件,它用 PGP 来保障电子邮件的安全。

在这种情况下,E-mail 客户端通过扩展,增加了下面这些额外的功能:

(1) 可在一个本地数据库里查找与某位特定用户对应的公共密钥。

(2) 可提供多种安全服务,比如加密/解密、不可抵赖(信件加上了自己的签名,其发件人不容抵赖)以及对电子邮件发件人的身份进行验证等。

对应用而言,应根据需要设计好自己的安全机制,不可依赖较低的层来提供这些服务。"不可抵赖"安全服务便是这样的一个例子。对低层服务来说,其实很难提供"不可抵赖"服务,因为它们没有权力访问数据。

下面简单介绍几种运行于应用层上的安全协议:

(1) SSH(secure shell),是一种远程登录的安全协议。它通过在终端机上将所有需要传输的数据进行加密,以此来防止"中间人"这样的攻击方式和 DNS、IP 欺骗等。其加密方式一般基于 RSA、Diffie-Helman 和数字签名等算法。

(2) SET(secure electronic transaction),是一个基于可信的第三方认证中心并以此实现网上交易的模型和规范。从概念上讲,它也是通用信用卡的自然延拓。SET 能够确保身份认证的安全可靠,实现信息的保密性和完整性要求。在 SET 中,通常使用的是 DES 与 RSA 混合的方法,目前也采用 AES 和 ECC 相结合的方法。

(3) S-HTTP(secure hypertext transfer protocol),是一个 Web 安全协议。它通过把加密增强功能集成到 HTTP 通信流中,在应用层实现对 WWW 的安全支持。该协议利用 MIME,基于文本进行加密、报文认证和密钥分发等。详细内容可参考第 6 章中相关内容。

(4) PEM(privacy enhanced mail),是 Internet 上最初的加密邮件协议。它主要实现包括报文加密、数据源认证以及确保消息完整性和不可抵赖性等功能。PEM 基于 RSA 和 DES 的密码算法来实现其安全性。同时,也可通过 ECC 和 AES 的配合使用来达到其安全性要求。

(5) S/MIME(secure/multipurpose internet mail extension),是一个电子邮件安全协议。通过使用签名、加密或签名/加密的组合来保证 MIME 通信的安全,并强化了证书的规范性。它主要采用 DES 和 RSA 密码算法,更加高级的应用是通过 AES 和 ECC 加密算法来实现。

## 2.3.3　传送层

与应用层安全相比，在传送层提供安全服务具有一些明显的好处，因为它不会强制要求每个应用都在安全方面作出相应的改进。即使现有的应用本身没有提供安全服务，也能自然、“无缝”地获得安全服务。

然而，由于要取得用户场景，所以局面也变得复杂起来。为提供由具体用户决定的服务，假定只有一名用户使用系统，而且这种假定目前已成为一种标准的做法。与应用层的安全类似，传送层的安全只可在端系统(end system)实现。

具体的传送层安全措施要取决于具体的协议。其中，名字叫“传送层安全”(TLS)的一种协议在 TCP 的顶部提供了如身份验证、完整性检验以及机密性保证这样的安全服务。TLS 需要为一个连接维持相应的场景，而且目前并未在 UDP 上实现，因为 UDP 并不维持任何场景。

由于安全机制与特定的传送协议有关，所以像密钥管理这样的安全服务可能使每种传送协议重复。

万维网(WWW)目前用 TLS 提供安全服务。然而，假如安全服务在网络层实现，它便会向下移至网络层。根据目前的定义，传送层安全的另一个限制是应用程序需要进行修改，才能请求传送层提供安全服务。

下面简单介绍几种运行于传送层上的安全协议：

(1) SSLP(secure sockets layer protocol)，是一个用于在 Internet 上进行保密通信的安全协议。准确地说，它是位于 TCP 层和应用层之间的，且对应用层透明。SSLP 所基于的是不同密钥长度的 RC-2 或 RC-4、DES 以及 AES、ECC 等高级密码算法和杂凑算法 MD5 的各种搭配。其更多详细内容在第 6 章中会讲到。

(2) TLS(transport layer security)，其实质是将 SSL 通用化的安全协议。它提供两个通信应用程序之间的保密性和数据完整性。该协议基于 DES、SHA、RSA 以及 AES、ECC 等密码算法。

(3) SOCKSv5，是防火墙及 VPN 用的数据加密及认证的安全协议。它提供网络应用的基于代理的服务，能够对所有 IP 应用进行认证和加密，同时还增加了强有力的安全协商机制。具体内容见第 6 章中相关小节。

## 2.3.4　网络层

在网络层实现安全服务具有多方面的优点。首先，密钥协商的开销被大大地削减了。这是由于多种传送协议和应用程序可共享由网络层提供的密钥管理架构。其次，假如安全服务在较低层实现，那么需要改动的应用程序便要少得多。通过它，我们不必集中在较高层实现大量安全协议。假如安全协议在较高层实现，那么每个应用都必须设计自己的安全机制。

这样做除极易产生安全漏洞以外，而且出现犯错误的概率也会增加。然而，对于任何传送协议，都可为其“无缝”地提供安全保障。

网络层安全最有用的一项特性是能够构建 VPN 和内部网(Intranet)。由于 VPN 和

内部网是以子网为基础,而且网络层支持以子网为基础的安全,所以很容易实现 VPN 和内部网。

在网络层提供安全服务的缺点是很难解决像数据的“不可抵赖”之类的问题。这样的问题最好还是在较高层解决。若在网络层提供安全服务,很难在一部多用户的机器上实现逐用户的控制。然而,可在终端主机上提供相应的机制,实现以用户为基础的安全保障。

在路由器上,由于不存在用户场景,所以这个问题不会出现。

作为本书的重点,IP 安全机制在网络层提供了安全服务。IPSec(Internet Protocol Security)是目前唯一一种能为任何形式的 Internet 通信提供安全保障的协议。此外,IPSec 也允许提供逐个数据流或者逐个连接的安全,所以能实现非常细致的安全控制。

IPSec,作为网络层上唯一使用的标准安全协议,在这里只是对其进行比较简单的介绍,详细的内容将在第 2.4 节中作进一步的阐述。

### 2.3.5 数据链路层

假定两个主机或路由器之间存在一条专用通信链路,而且为避免有人“窥视”,所有通信都需加密,可用硬件设备来进行数据加密。

这样做最大的好处在于速度。然而,该方案不易扩展,而且仅在专用链路上才能很好地工作。另外,进行通信的两个实体必须在物理上连接到一起。

这种安全模型在自动柜员机(ATM)上得到了广泛的应用。所有机器均通过专用线路连接到中心办公室。假如 ATM 连接到一个 IP 网络,而不是采用专用的安全链路,那么数据链路层的安全并不足以保证通信的安全,必须向上移动一层,以提供安全服务。

下面简单介绍几种运行于数据链路层上的安全协议:

(1) PPTP(point to point tunneling protocol),是以 IP 协议封装 PPP 帧,通过在 IP 网上建立隧道来透明传送 PPP 帧的隧道协议。通过该协议,远程用户能够通过 Microsoft Windows NT 工作站和装有点对点协议的系统安全访问公司网络;并能拨号连入本地 ISP,通过 Internet 安全链接到公司网络。

(2) L2F(layer2 forwarding),是一种允许高层协议的链路层隧道技术,用于建立跨越公共网络(如因特网)的安全隧道来将 ISP POP 连接到企业内部网关。

(3) L2TP(layer2 tunneling protocol),是把链路层 PPP 帧封装在公共网络设施如 IP、ATM、帧中继中进行隧道传输的封装协议,它结合了 PPTP 和 L2F 的优点,可以使用户从客户端或访问服务器端发起 VPN 连接。

## 2.4 网络层的安全协议 IPSec

Internet 网络的重要特点是以 IP 协议作为其网络层的唯一标准协议,任何借助 Internet 网络的通信,在网络层都必须使用 IP 协议。因此,如果能够保证 IP 层的安全,那么,在很大程度上就保证了 IP 网上的通信安全,这便是 IETF(Internet Engineering

Task Force)网络安全组制定 IPSec 安全协议的基本思路。而 IP 安全协议正是解决 IP 通信安全的一个可行的解决方案。

因为 IP 包本身并不继承任何安全特性,很容易便可伪造出 IP 包的地址、修改其内容、重播以前的包以及在传输途中拦截并查看包的内容。因此,我们不能担保收到的 IP 数据报下述情况是否成立：

(1) 来自原先要求的发送方(IP 头内的源地址)。

(2) 包含的是发送方当初放在其中的原始数据。

(3) 原始数据在传输途中未被其他人看过。

针对这些问题,IPSec 可有效地保护 IP 数据报的安全。它采取的具体保护形式包括：数据起源地验证；无连接数据的完整性验证；数据内容的机密性(是否被别人看过)；抗重播保护以及有限的数据流机密性保证。

IPSec 是 IETF IPSec 工作组以 RFC 形式公布的一组安全 IP 协议集,是在 IP 包级为 IP 业务提供保护的安全协议标准,其基本目的是把密码学的安全机制引入 IP 协议,通过使用现代密码学方法支持保密和认证服务,使用户能有选择地使用,并得到所期望的安全服务。IPSec 是随着 IPv6 的制定而产生的,鉴于 IPv4 的应用仍然很广泛,所以后来在 IPSec 的制定中也增加了对 IPv4 的支持。IPSec 在 IPv6 中是必须支持的,而在 IPv4 中是可选的。IPSec 是一个网络安全协议工业标准,它主要由 3 个基本部分组成：认证头部 AH(authentication header)、封装安全净荷 ESP(encapsulating security payload)以及密钥交换(IKE)。

## 2.4.1　IPSec 的体系结构

IPSec 将几种安全技术结合形成一个完整的安全体系,它包括安全协议部分和密钥协商部分。在协议描述上,IPSec 按照其框架设计,主要用了 8 个 RFC 文档来定义 IPSec 协议。图 2-6 显示了 IPSec 的体系结构、组件及各组件间的相互关系。

(1) RFC2401：IPSec 的体系文档。定义了 IPSec 的基本结构,指定 IP 包的机密性(加密)和身份认证使用传输安全协议 ESP 或 AH 实现,包含了一般的概念、安全需求、定义和定义 IPSec 的技术机制。

(2) RFC2406：协议文档 ESP(encapsulate security payload,封装安全载荷)。定义了为通信提供机密性、完整性保护和抗重放服务的具体实现方法,以及 ESP 头在 ESP 实现中应插入 IP 头的位置、ESP 载荷格式、各字段的语义、取值方式以及对进入分组和外出分组的处理过程等。

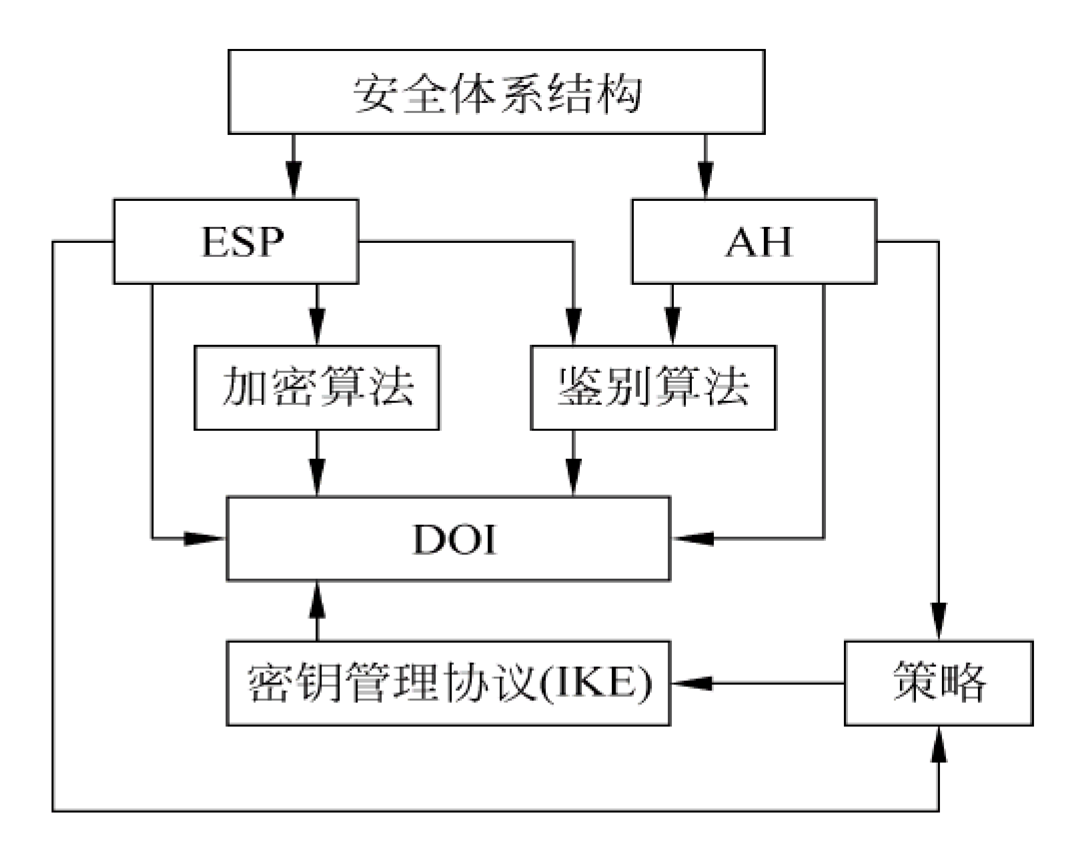



图 2-6　IPSec 的体系结构

(3) RFC2402：协议文档认证头(authentication header,AH)。定义了为通信提供完整性和抗重放服务的具体实现方法,以及 AH 头在 AH 实现中应插入 IP 头的位置、AH 头的语法格式、各字段的语义、取值方式以及实施 AH 时对进入分组和外出分组的处理

过程。

（4）RFC2403 和 RFC2404：认证算法。RFC2403 定义了对 ESP 的认证算法为散列函数 MC5 或 SHA 的 HMAC 版本，RFC2404 定义了默认情况下 AH 的认证算法是 MD5 或 SHA 的 HAMC 版本。

（5）RFC2405：加密算法。定义了 DES-CBC 作为 ESP 的加密算法以及如何实现 DES-CBC 算法和初始化矢量（IV）的生成。

（6）RFC2409：因特网密钥交换（Internet key exchange，IKE）。定义了 IPSec 通信双方如何动态建立共享安全参数和经认证过的密钥（即建立安全关联），IKE 的功能包括：加密/鉴别算法和密钥的协商、密钥生成、交换及管理、通信的保护模式（传输或隧道模式）、密钥的生存期等。

（7）RFC2407：IPSec 解释域（domain of interpretation，DOI）。IKE 定义了安全参数如何协商以及共享密钥如何建立，但没有定义协商内容（参数）。协商内容与 IKE 协议本身分开实现。协商的内容（参数）被归于一个单独的文档内，名为 IPSec DOI。

（8）策略：策略是两个实体间通信的规则，它决定采用什么协议、什么加密算法和认证算法来通信，策略不当可能造成不能正常工作。目前，策略还没有统一标准。

## 2.4.2 安全关联和安全策略

### 1. 安全关联（SA）

安全关联（security association，SA）是构成 IPSec 的基础，是两个通信实体经协商建立起来的一种协定，它们决定了用来保护数据包安全的安全协议（AH 协议或者 ESP 协议）、转码方式、密钥及密钥的有效存在时间等。SA 通过一个三元组唯一的标识。该三元组包括：

（1）安全参数索引（security parameters index，SPI），SPI 是安全关联 SA 与 IPSec 安全协议的唯一接口。安全协议通过 SPI 查找对应的 SA，并使用 SA 中的对应安全参数对通信进行保护。

（2）源或目的 IP 地址：表示对方 IP 地址。对于外出数据包，指目的 IP 地址；对于进入数据包，指源 IP 地址。

（3）一个特定的 IPSec 协议：采用 AH 协议或者 ESP 协议。

随每个数据包一道，都要发送一个 SPI，以便将 SA 独一无二地标识出来。SA 是单向的，如果两个主机 A 和 B 正在通过 ESP 进行安全通信，则主机 A 需要一个 SA（out）来处理数据报，另外还需要一个 SA（in）用来处理进入 A 的数据包。

SA 是通过像 IKE 这样的密钥管理协议在通信双方之间协商的。当一个 SA 的协商完成时，通信双方都在它们的安全关联数据库（SAD）中存储该 SA 参数。

### 2. 安全关联数据库（SAD）

安全关联数据库（security association database，SAD）并不是通常意义上的“数据库”，而是将所有的 SA 以某种数据结构集中存储的一个列表。

SAD 中包含现行的 SA 条目，每个 SA 包括一个三元组，用于唯一地标识这个 SA，该三元组包含一个 SPI，一个用于输入处理 SA 的源 IP 地址，另一个用于输出处理 SA 的目的 IP 地址和一特定的协议（例如 AH 或者 ESP）。此外，一个 SAD 条目包含以下的域：序列号计数器、序列号溢出标志、抗重放窗口、AH 认证密码算法和所需要的密钥、ESP 认证密码算法和所需要的密钥、ESP 加密算法、密钥、初始化向量（IV）和 IV 模式、IPSec 协议操作模式、路径最大传输单元（PMTU）和 SA 生存期。IPSec 处理对于输入和输出要保存单独的 SAD。对于输入或者输出通信，将搜索各自的 SAD 来查找与从数据包头域中解析出来的与选择符相匹配的 SPI、源或者目的地址以及 IPSec 协议。

如果找到一个匹配的条目，则将该 SA 的参数与 AH 或 ESP 头中的适当域相比较。如果头域与 SAD 中的 SA 参数一致，就处理该数据包。如果有任何差别，就丢弃该包。如果没有 SA 条目与选择符相匹配，并且数据包是一个输入包，就将它丢弃；如果数据包是输出的，则创建一个新的 SA 或者 SA 束，并将其存入输出 SAD 中。

#### 3. 安全策略（SP）

安全策略（security policy，SP）是 IPSec 结构中非常重要的组件，它定义了两个实体之间的安全通信。指示是否对 IP 数据包进行安全保护，以及使用什么样的安全保护。SP 主要根据源 IP 地址、目的 IP 地址、输入数据还是输出数据等来标识。

IPSec 还定义了用户能以何种粒度来设定自己的安全策略，由“选择符”来控制粒度的大小，不仅可以控制到 IP 地址，还可以控制到传输层或者 TCP/UDP 端口等。

#### 4. 安全策略数据库

安全策略数据库（security policy database，SPD）也不是通常意义上的“数据库”，而是将所有的 SP 以某种数据结构集中存储的一个列表。

通常，IPSec 协议要求在所有数据包（通信流）进行处理的过程中都必须查询 SPD，不管是输入的还是输出的。SPD 中包含一个策略条目的有序列表。通过使用一个或者多个选择项来确定每一个条目。IPSec 当前允许的选择项有：目的 IP 地址、源 IP 地址、传输层协议、系统名和用户 ID。SPD 中的每一个条目都包含一个或者多个选择符和一个标志，该标志用于表明与条目中的选择符匹配的数据包是应该丢弃、绕过还是进行 IPSec 处理。如果应该对数据包进行 IPSec 处理，则条目中必须包含一个指向 SA 内容的指针，其中详细说明了应用于匹配该策略条目的数据包的 IPSec 协议、操作模式以及密码算法。

### 2.4.3　IPSec 协议的运行模式

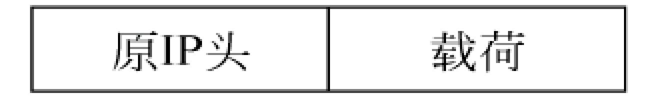
#### 1. IPSec 隧道模式

这种模式的特点是数据包最终目的地不是安全终点。通常情况下，只要 IPSec 双方有一方是安全网关或路由器，就必须使用隧道模式。在隧道模式中，IPSec 对整个 IP 包进行封装保护，并增加一个新的外部 IP 头，同时在外部与内部 IP 头之间插入一个 IPSec 头，如图 2-7 所示。

### 2. IPSec 传输模式

传输模式下，IPSec 主要对上层协议即 IP 包的载荷进行封装保护，通常情况下，传输模式只用于两台主机之间的安全通信。在传输模式中，原 IP 头保持不变，IP 头与上层协议之间需要插入一个特殊的 IPSec 头，如图 2-7 所示。

原始IP数据包




隧道模式下受IPSec保护的IP数据包

| 新IP头 | IPSec头 | 原IP头 | 载荷 |
| --- | --- | --- | --- |

传输模式下受IPSec保护的IP数据包

| 原IP头 | IPSec头 | 载荷 |
| --- | --- | --- |

图 2-7　隧道模式和传输模式下受 IPSec 保护的数据包格式

## 2.4.4　AH 协议

设计 AH 认证协议的目的是用来增加 IP 数据报的安全性。AH 协议提供无连接的完整性、数据源认证和抗重放保护服务，但是，AH 不提供任何保密性服务。AH 的作用是为 IP 数据流提供高强度的密码认证，以确保被修改过的数据包可以被检查出来。AH 用带密钥的 HMAC 算法和迭代型杂凑函数（如 MD5、SHA-1）结合使用，而不用对杂凑函数进行修改。

### 2.4.4.1　认证头格式

AH 由 5 个固定长度域和 1 个变长的认证数据域组成。图 2-8 说明了这些域在 AH 中的位置。

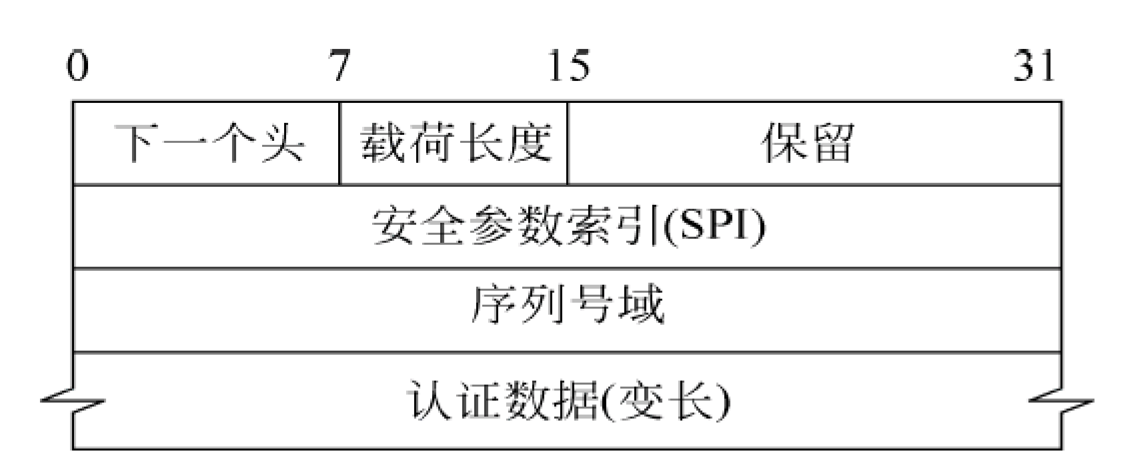



图 2-8　IPSec 认证头的格式

图 2-8 中有关域的描述如下。

(1) 下一个头(next header)：这个 8 比特的域指出 AH 后的下一载荷的类型。例如，如果 AH 后面是一个 ESP 载荷，这个域将包含值 50。

(2) 载荷长度(payload length)：这个 8 比特的域包含以 32 比特为单位的 AH 的长

度减 2。为什么减 2 呢？AH 实际上是一个 IPv6 扩展头，IPv6 规范 RFC1883 规定计算扩展头长度时应首先从头长度中减去一个 64 比特的字。由于载荷长度单位用 32 比特度量，两个 32 比特字也就相当于一个 64 比特字，因此要从总认证头长度中减去 2。

（3）保留（reserved）：这个 16 比特的保留域供将来使用。AH 规范 RFC2402 规定这个域被置为 0。

（4）安全参数索引（SPI）：SPI 是一个 32 比特的整数，用于和源地址或目的地址以及 IPSec 协议（AH 或 ESP）共同唯一标识一个数据报所属的数据流的安全关联（SA）。关于 SPI 域的整数值，1～255 被 IANA 留做将来使用；0 被保留，用于本地和具体实现。所以，目前有效的 SPI 值从 256～($2^{32}-1$)。

（5）序列号（sequence number）：这个域包含一个作为单调增加计数器的 32 位无符号整数。当 SA 建立时，发送者和接收者的序列号值均被初始化为 0。通信双方使用一个特定的 SA 发出一个数据报就将它们的相应序列号加 1。序列号用来防止对数据包的重放，采用滑动接收窗口机制检测重放包。重放是指数据包被攻击者截取并重新传送。

（6）认证数据：这个变长域包含数据包的认证数据，该认证数据被称为数据包的完整性校验值（ICV）。对于 IPv4 数据包，这个域的长度必须是 32 的整数倍；对于 IPv6 数据包，这个域的长度必须是 64 的整数倍。用来生成 ICV 的算法由 SA 指定。为了保证互操作性，AH 强制所有的 IPSec 实现必须包含两个认证算法：HMAC-MD5-96 和 HMAC-SHA-1-96。如果一个 IPv4 数据包的 ICV 域的长度不是 32 的整数倍，或一个 IPv6 数据包的 ICV 域的长度不是 64 的整数倍，必须添加填充比特使 ICV 域的长度达到所需要的长度。

### 2.4.4.2　AH 的操作模式

按照 AH 协议的规定，可以按 AH 封装的协议数据不同，将 AH 封装分为两种模式：传输模式和隧道模式，AH 头的位置依赖于 AH 的操作模式。使用隧道模式时，AH 可以以一种嵌套方式使用，也可以和 ESP 组合使用或单独使用。

#### 1. AH 传输模式

在传输模式中，AH 被插在 IP 头之后但在所有的传输层协议之前，或所有其他 IPSec 协议头之前。图 2-9 说明了在传输模式中的 AH 封装格式。

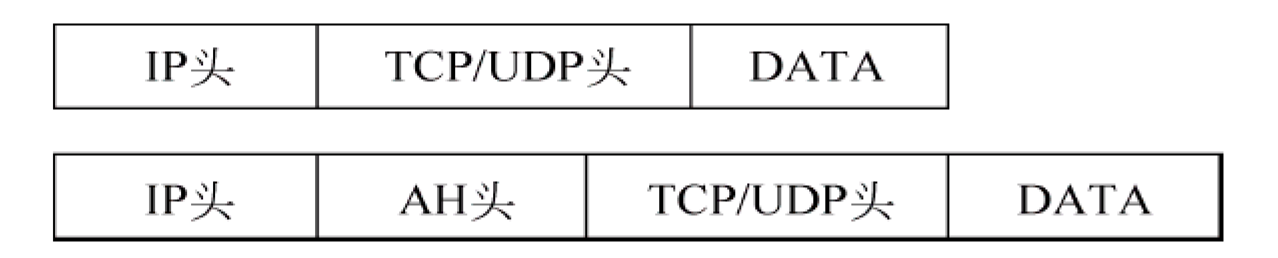



图 2-9　传输模式下 AH 的封装

与 ESP 认证范围不同，AH 认证整个 IP 数据报，包括 IP 包头部，因此，源 IP 地址、目的 IP 地址是不能修改的，否则会被检测出来。所以，AH 用于传输模式有其局限性。例如一个内部网中的主机 A 有一个私有 IP 地址（私有地址在公网上是不可路由的），通过 Internet 与另一台主机 B 进行通信，要求通过 AH 传输模式进行数据完整性保护。主机

A 的数据流在离开源网络与位于它们所在网络之外的主机 B 进行通信前，需要经过一个网络地址转换（NAT）网关，NAT 将流出的数据包的源地址域的私有 IP 地址替换为指定的公网 IP 地址，并将数据包转发给指定的目的地址。这样，主机 B 计算出来的完整性校验值将和主机 A 计算出来的不同，因为 NAT 网关修改了数据包的源地址域。AH 完整性认证失败。

如果希望为上面的例子提供 AH 传输模式认证，认证需要在网关上进行。但这个解决方案对数据流提供的保护是不完整的：主机到网关和网关到主机之间的数据流没有受到保护。我们可以采用 ESP 提供额外的安全性。

### 2. AH 隧道模式

在隧道模式中，AH 插在原始的 IP 头之前，另外生成一个新的 IP 头放在 AH 之前。图 2-10 说明了在隧道模式中的 AH 封装格式。




图 2-10 隧道模式下 AH 的封装

同在传输模式下一样，AH 验证整个 IP 数据报，包括新的 IP 头，前面讨论过 AH 在传输模式下的局限性在隧道模式下同样存在。如果希望对位于安全网关后或在 NAT 环境中的通信双方进行 AH 隧道模式认证的话，那么验证应该在网关而不是在通信主机上进行。

## 2.4.4.3 对数据包的 AH 处理

### 1. 对输出数据包的处理

（1）查找对应的 SA。根据待处理的数据包，构造出选择符（源 IP 地址、目的 IP 地址、协议号、通信端口），并以此选择符为索引检查 SPD。假如按检索到的 SPD 条目，该分组应受到 AH 保护，则按 SPD 条目直接指向的 SA 条目找到用以处理数据的 SA。如果 SPD 指向的 SA 为空，则采用 IKE 协议协商新的 SA。

（2）按 SA 条目给出的处理模式，在适当的地方插入 AH 头和外部 IP 头。在传输模式的 AH 实现中，待添加的 IP 头为原 IP 头；对于隧道模式，将重新构造新的 IP 头，添加到 AH 载荷前面。

（3）对 AH 载荷的相应字段进行填充。下一载荷头来源于原 IP 头的协议字段（对传输模式）或为 4（对隧道模式，表示 IP 协议头）；载荷长度字段值将 AH 载荷的总字数（每字 32 位）减 2；保留字段填充 0；安全参数索引 SA 条目；序列号来源于 IPSec 处理的发送端 SA 计数器；验证字段的值根据 SA 中指定的认证算法对载荷数据进行散列计算得到。需要注意的是，在计算验证字段时，应把可能在传输过程中改变的值，或不可能知道到达隧道对端时的预期值，都应先置 0。

(4) 对 AH 处理后的数据包,重新计算 IP 头校验和。如果处理后的数据包的长度大于本地 MTU,则进行 IP 分段。处理完毕的 IPSec 分组被交给数据链路层(对传输模式)或 IP 层(对隧道模式)重新路由。

#### 2. 对输入数据包的处理

(1) 数据包重组：对于收到的数据包,查看一下是否是一个完整的受 AH 保护的数据包,如果得到的数据包仅是一个 IP 包的分段,则必须保留这个分段,直到收到属于该数据包的所有分段,并成功重组之后,再进行后续处理。

(2) 查找 SA(目的 IP 地址、协议号 SPI)：三元组一般被称为 SAID,即该三元组被用于标识一个 SA。由待处理的 IPSec 数据包提取 SAID,利用它作为索引对 SADB 进行检索,找到相应的处理该数据包的 SA。如果没有找到,则丢弃该数据包,并将此事件记录于日志中。

(3) 序列号检查：如果此数据包的序列号落在该活动 SA 的滑动窗口内,且不是一个重复收到的数据包,则表明此数据包是有效的,继续后续处理；否则丢弃该数据包,并将此事件记录于日志中。

(4) 检查 ICV 字段的值：首先将 AH 头中的 ICV 字段保存下来,然后将这个字段清零,按鉴别算法和鉴别密钥,与发送端相同的计算方式计算一个散列输出。该散列输出与保存的 ICV 相比较,如果相同,则通过完整性检查；否则,丢弃该数据包,并将此事件记录于日志中。

(5) 重构明文数据包：对传输模式的 AH 封装而言,将 AH 载荷的"下一载荷头"字段值赋予 IP 头的协议字段；去除 AH 封装(即去掉包括 SPI、序列号、填充、填充长度、下一载荷头以及 ICV 字段),对得到的 IP 数据包重新计算 IP 头校验和。对于隧道模式的 AH 封装,内部 IP 头即是原 IP 头。因此,恢复处理只需要去掉外部 IP 头、AH 头,便可得到原明文 IP 数据包。

(6) 对处理的合法性检查：ICV 校验完成后,应将该数据包的保护方式与 SA 指向的 SPD 条目的安全策略相比较,以检验安全实施的一致性。若相符,则继续后续处理；否则丢弃该数据包,并将此事件记录于日志中。

## 2.4.5　ESP 协议

封装安全载荷(ESP)用于提高 Internet 协议(IP 协议)的安全性。它可为 IP 提供机密性、数据源验证、抗重放以及数据完整性等安全服务。ESP 中用来加密数据的密码算法都毫无例外地使用对称密码体制。协议规定随 ESP 使用的所有加密算法必须以"加密块链接(CBC)"模式工作。

### 2.4.5.1　ESP 数据包格式

ESP 数据包由 4 个固定长度的域和 3 个变长域组成。这个协议的包格式如图 2-11 所示。

图 2-11 中有关域的描述如下：

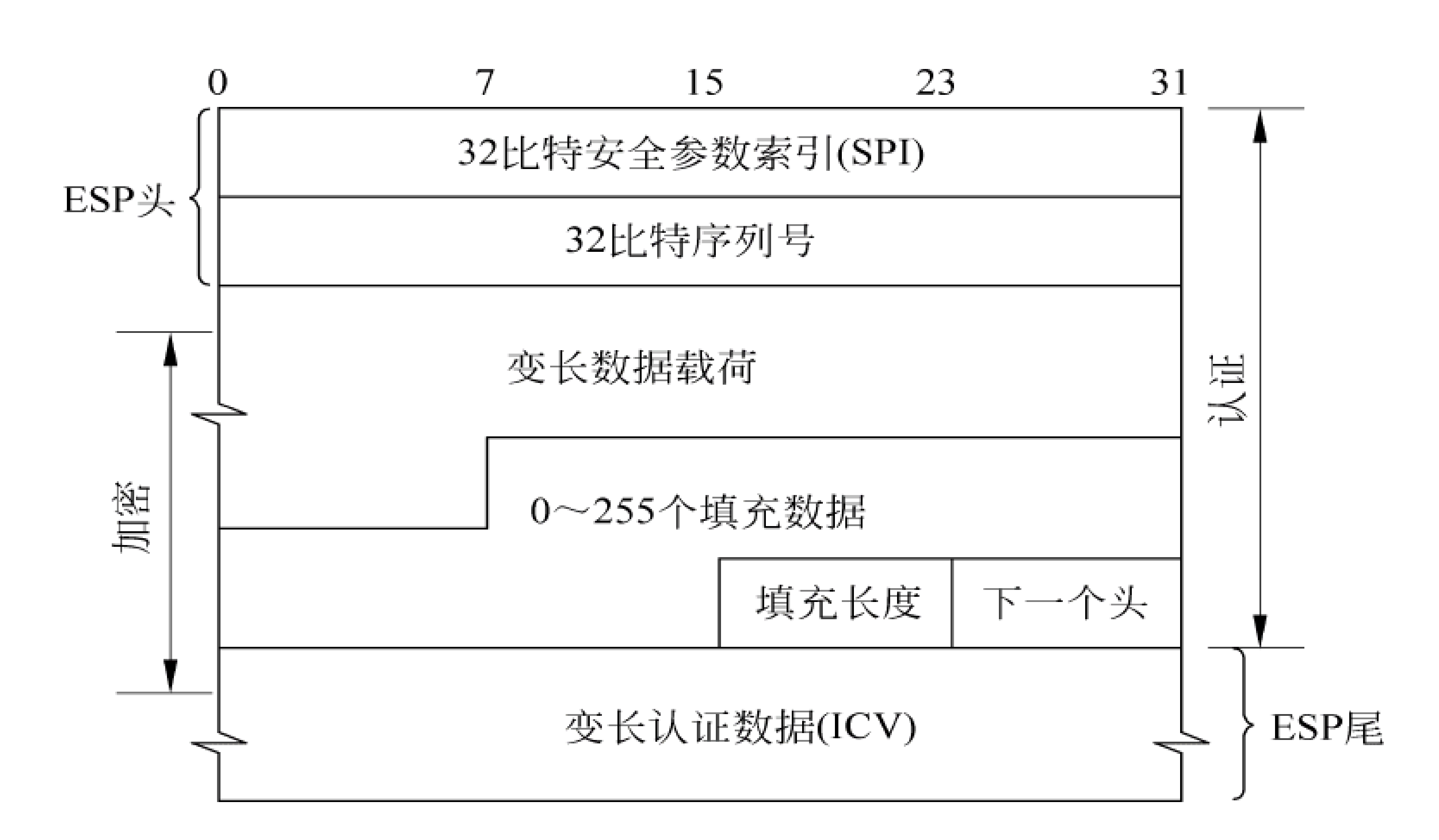



图 2-11　ESP 数据包格式

(1) 安全参数索引(SPI)：这个域和 AH 的 SPI 类似。

(2) 序列号(sequence number)：和 AH 的情况类似。

(3) 载荷数据(payload data)：这是一个变长域。如果使用保密服务，其中就包含实际的载荷数据(即数据报加密部分的密文)。这个域是必须有的，不管涉及的 SA 是否需要保密服务。如果采用的加密算法需要初始化向量(IV)，它将在载荷域中传输，并且算法的规范要指明 IV 的长度和它在载荷数据域中的位置。载荷数据域的长度以比特为单位并且必须是 8 的整数倍。

(4) 填充(padding)：如果有的话，这个域包含填充比特，由加密算法使用或用于使填充长度域和 4 字节字中的第 3 个字节对齐(见图 2-11 所示)。这个域的有效值是 0～255 之间的整数。

(5) 填充长度(pad length)：填充长度是一个 8 比特的域，这个值是必需的，表明填充域中填充比特的长度。这个域的有效值是 0～255 之间的整数。

(6) 下一个头(next header)：这个 8 比特的域表明载荷中封装的数据类型。可能是一个 IPv6 扩展头或传输层协议。例如，值 6 表明载荷中封装的是 TCP 数据。

(7) 认证数据(authentication data)：这个变长域中存放完整性检查值(ICV)，如图 2-11 所示。它是对除认证数据域外的 ESP 包进行计算获得的。这个域的实际长度取决于使用的认证算法。例如，如果使用 HMAC-MD5 则认证数据域是 128 比特；如果取其前 96 位，则认证数据域是 96 比特。认证数据域是可选的，仅当指定的 SA 要求 ESP 认证服务时才包含它。

### 2.4.5.2　ESP 的操作模式

和 AH 的情况一样，ESP 在数据包中的位置取决于 ESP 的操作模式。ESP 共有两种操作模式：传输模式和隧道模式。

#### 1. 传输模式

在传输模式下，ESP 被插在 IP 头和所有的选项之后，但是在传输层协议之前，或者在

已应用的任意 IPSec 协议之前。图 2-12 说明了在传输模式中的 ESP 封装格式。

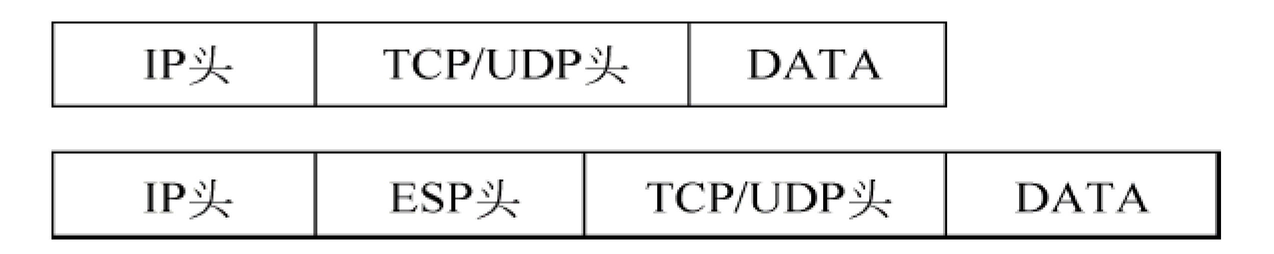



图 2-12　传输模式下的 ESP 封装格式

传输 ESP 认证服务与 AH 不同，ESP 不对整个 IP 数据包进行认证，所以，在传输模式下 ESP 不受 AH 传输模式下的局限性约束。使用私有 IP 地址(通过 Internet)或位于安全网关之后的主机间的通信可被 ESP 认证服务保护，因为 IP 头中的源和目的以及其他域未被认证。

ESP 提供的这种灵活性也显示出了它的弱点：首先，ESP 传输模式认证服务提供的安全性不如 AH 传输模式。因为除了 ESP 头部外，在从源到目的的传输过程中 IP 头的任何域都可以被修改，如果修改后头部校验和计算正确，目的主机将无法检测到发生的修改。其次，在传输模式下的 ESP 不提供数据流保密服务，因为源和目的 IP 地址未被加密。

### 2. 隧道模式

在隧道模式下，ESP 被插在原始 IP 头之前，并且生成一个新的 IP 将其插在 ESP 之前。内部 IP 头中包括真正的源地址和最终的目的地址。外部的源和目的 IP 头域分别包含源及目的节点的安全网关的地址。图 2-13 说明了在隧道模式中的 ESP 封装格式。

| | | |
|---|---|---|
| IP头 | TCP/UDP头 | DATA |

| | | | | |
|---|---|---|---|---|
| 新IP头 | ESP头 | 原IP头 | TCP/UDP头 | DATA |

图 2-13　隧道模式下的 ESP 封装格式

隧道模式的认证和保密服务要对整个内部 IP 头进行认证和加密。而外部头既未被认证也未被加密，它未被加密是因为路由器需要其中的信息对数据包进行路由导向；未被认证是因为如果认证的话会导致像 AH 认证服务的局限性。

## 2.4.5.3　对数据包的 ESP 处理

### 1. 对输出数据包的处理

(1) 查找对应的 SA：根据待处理的数据包，构造出选择符(源 IP 地址、目的 IP 地址、协议号、通信端口)，并以此选择符为索引检查 SPD。假如按检索到的 SPD 条目，该分组应受到 AH 保护，则按 SPD 条目直接指向的 SA 条目找到用以处理数据的 SA。如果 SPD 指向的 SA 为空，则采用 IKE 协议协商新的 SA。

(2) 构造 ESP 载荷：在传输模式下，IP 头的协议字段被复制到 ESP 头的下一载荷字段；SPI 字段来源于用来对此数据包进行处理的 SA 的 SPI 标识符值；序列号为 SA 计数

器的下一取值；按 DOI 的描述规则计算的初始向量被填入 ESP 载荷数据的起始部分(如果需要)；原 IP 数据包除去 IP 头或其他的扩展头后的部分紧接初始向量被填入载荷数据；依据载荷长度，按相应的填充规则，对载荷数据进行填充；填充的长度被填入填充长度字段。在隧道模式下，ESP 头被加入到原 IP 头之前，对 SPI、序列号、初始向量有与传输模式相同的填充方式；初始向量之后紧随整个原 IP 数据包；填充、填充长度字段也与传输模式的填充方式一致；下一载荷头的取值则依据 IANA 在“已分配协议号”中的 IP 协议号集的规定，如果取值为 4，则表明 ESP 封装的是 IPv4 数据包；如果取值为 41，则表明 ESP 封装的是 IPv6 分组。

(3) 为 ESP 载荷添加 IP 头：在传输模式的 ESP 实现中，待添加的 IP 头为原 IP 头；对于隧道模式，将重新构造新的 IP 头，添加到 ESP 载荷前面。

(4) 对 ESP 载荷进行加密及鉴别处理：SA 的处理要求，如果协商时要求了加密保护，则利用适当的加密算法和加密密钥(都由 SA 给出)，对起于初始向量之后、止于下一载荷头字段之间的数据进行加密处理，并以输出的密文代替原明文部分；如果还启动了鉴别功能，则利用 SA 中给出的鉴别算法和鉴别密钥对整个 ESP 载荷(自 ESP 头开始，包括密文部分，一直到 ESP 尾，包括全部置 0 的 ICV 字段在内)进行散列计算，并将结果填入 ESP 的 ICV 字段。

(5) 其他处理：对输出数据包重新计算外部 IP 头的校验和。如果封装后的 IP 分组的长度超过本地的 MTU，则应对其进行分段处理。

### 2. 对输入数据包的处理

(1) 数据包重组：对于收到的数据包，查看一下是否是一个完整的受 ESP 保护的数据包，如果得到的数据包仅是一个 IP 包的分段，则必须保留这个分段，直到收到属于该数据包的所有分段，并成功重组之后，再进行后续处理。

(2) 由待处理的 IPSec 数据包提取 SAID(即目的 IP 地址、协议号 SPI)：利用它作为索引对 SAD 进行检索，找到相应的处理该数据包的 SA。如果没有找到，则丢弃该数据包，并将此事件记录于日志中。

(3) 序列号检查：如果此数据包的序列号落在该活动 SA 的滑动窗口内，且不是一个重复收到的数据包，则表明此数据包是有效的，继续后续处理；否则丢弃该数据包，并将此事件记录于日志中。

(4) 完整性检验：对整个 ESP 载荷进行 ICV 重构，计算方法与发送端完全相同。鉴别密钥来源于检索到的 SA。本地计算出来的 ICV 与收到的数据包中的 ICV 进行一致性比较，若一致，该数据包通过完整性检验，进入下一步处理；否则丢弃该数据包，并将此事件记录于日志中。

(5) 数据包解密：利用 SA 中提供的密码算法和解密密钥，对 ESP 载荷的数据部分进行解密，受解密处理的范围包括初始向量之后，直到下一载荷的全部数据。若解密成功，则继续后续处理；否则丢弃该数据包，并将此事件记录于日志中。

(6) 重构明文数据包：对传输模式的 ESP 封装而言，将 ESP 载荷的“下一载荷头”字段值赋予 IP 头的协议字段；去除 ESP 封装(即去掉包括 SPI、序列号、IV、填充、填充长

度、下一载荷头以及 ICV 字段)；对得到的 IP 数据包重新计算 IP 头校验和。对于隧道模式的 ESP 封装,内部 IP 头即是原 IP 头。因此,恢复处理只需要去掉外部 IP 头、ESP 头、ESP 尾,便可得到原明文 IP 数据包。

(7) 对处理的合法性检查：完整性检查与解密完成后,应将该数据包的保护方式与 SA 指向的 SPD 条目的安全策略相比较,以检验安全实施的一致性。

## 2.4.6　Internet 密钥交换协议

Internet 密钥交换协议(Internet key exchange,IKE)是 IPSec 默认的安全密钥协商方法。IKE 通过一系列报文交换为两个实体(如网络终端或网关)进行安全通信派生会话密钥。这些交换为一些报文提供加密或认证保护,以及不同程度的防泛滥、重放和欺骗等攻击。IKE 主要依赖于很强的安全机制：公钥、私钥加密以及加密的哈希函数等,同时也依赖于公钥基础设施(PKI)。

### 2.4.6.1　IKE 协议的组成

IKE 是一种混合型的协议,它使用了 Internet 安全关联密钥管理协议(Internet security architecture and key management protocol,ISAKMP)的语言,综合了 Oakley 和 SKEME 的密钥交换方案,形成了独一无二的受鉴别保护的加密材料生成技术。

ISAKMP 提供了认证和密钥交换的框架,定义了一次特定的验证密钥交换完成的过程以及建立安全联盟所需的属性,但是并没有定义具体的密钥交换技术。

以 Oakley 为基础,IKE 借鉴了不同模式的思想,每种模式提供不同的服务,但都产生一个结果：通过验证的密钥交换。IKE 对这些模式进行了规范,将其定义成正规的密钥交换方法。尽管降低了 Oakley 模型的灵活性,IKE 仍然提供了“主模式”、“野蛮模式”和“快速模式”供用户选择。所以,最终还是成为一个非常适宜的密钥交换技术。

SKEME 定义了验证密钥交换的一种类型,其中,通信各方利用公共密钥加密实现相互间的验证,同时“共享”交换的组件。每一方都要用对方的公共密钥来加密一个随机数字,两个随机数字(解密后)都会对最终的密钥产生影响。IKE 在它的公共密钥加密验证中,直接借用了 SKEME 的这种技术；同时也借用了快速密钥刷新的概念。

### 2.4.6.2　IKE 的工作原理

IKE 协议定义了密钥协商的两个阶段和 Diffie-Hellman 密钥交换标准密码组：第一阶段在 ISAKMP 进程间建立一个安全并经过验证的信道,同时为 IPSecSA(第二阶段)产生密钥材料,即完成 ISAKMP 安全联盟(ISAKMPSA)；第二阶段指协商具体服务所需安全联盟,这些服务可以是 IPSec 或任何其他需要密钥材料协商参数的服务。

#### 1. IKE 第一阶段的协商

第一阶段通信的双方建立一个通过身份验证和安全保护的通道,即产生 ISAKMPSA。IKE 第一阶段提供了两种不同的模式：主模式和积极模式。在主模式下,发起者和响应者一共交换 6 条消息,并提供了身份验证和参数配置上的灵活性；积极模式只交换 3 条

消息，其最大的优点是速度快，但以降低协议的安全性为代价。在第一阶段中通信双方需要进行身份验证，这两种模式都支持 4 种不同的验证机制：预共享密钥的验证方法、公钥签名的验证方法、标准公钥加密的验证方法和改进的公钥加密的验证方法。

1）主模式

主模式在 3 个交换中总共用到了 6 条消息，最终建立了 ISAKMPSA。这 3 个步骤分别是模式协商、一次 Diffie-Hellman 交换和一次 Nonce 交换以及对对方身份的验证的交换。主模式的特点包括身份保护以及对 ISAKMP 协商能力的完全利用。其中，身份保护在对方希望隐藏自己的身份时显得尤为重要。以采用数字签名认证方法为例，主模式的交换过程如图 2-14 所示。

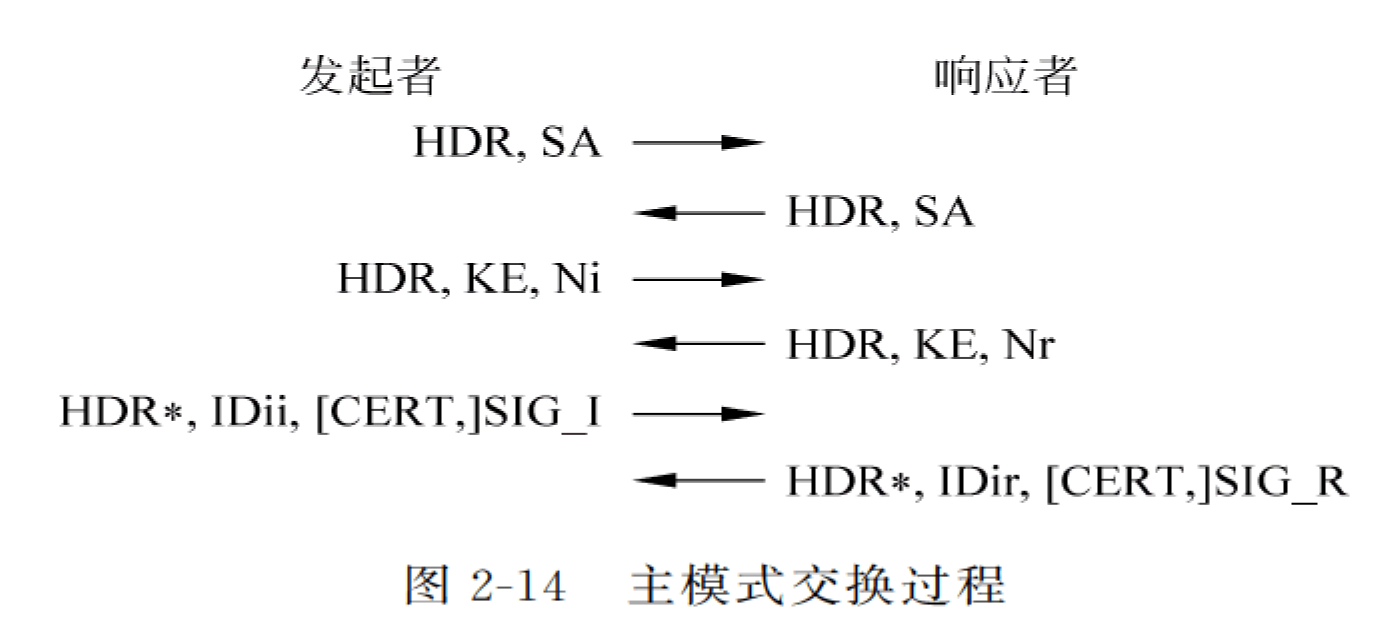
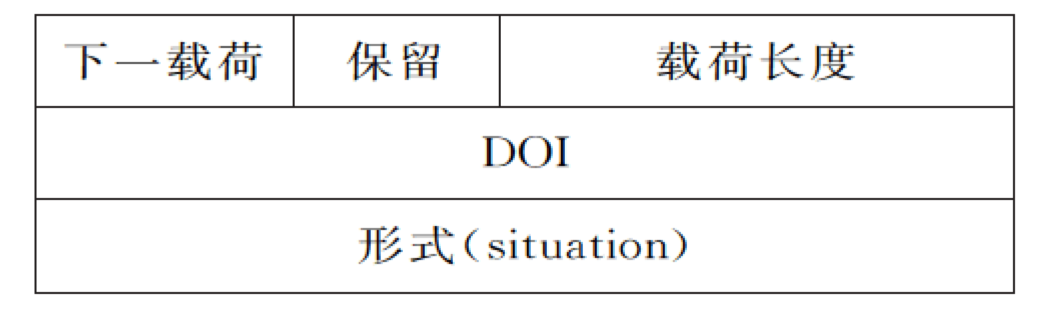



图 2-14　主模式交换过程

图中，HDR 表示 ISAKMP 通用头，所有的 ISAKMP 消息都以它开头，其格式如图 2-15 所示；HDR * 表示对载荷加密；SAi 和 SAr 表示协商的安全参数载荷，其格式如图 2-16 所示；Ni 和 Nr 表示 Nonce 载荷；IDii 和 IDir 分别表示第一阶段发起方和接收方的身份载荷；KE 表示 Diffie-Hellman 交换载荷；SIG_I 和 SIG_R 表示用于验证身份和消息的数字签名值，表示证书可选。

| | | | | |
|---|---|---|---|---|
| 发起者 Cookie | | | | |
| 响应者 Cookie | | | | |
| 下一载荷 | 主版本号 | 从版本号 | 交换类型 | 标志 |
| 消息 ID | | | | |
| 长度 | | | | |

图 2-15　ISAKMP 通用头



| | | |
|---|---|---|
| 下一载荷 | 保留 | 载荷长度 |
| DOI | | |
| 形式(situation) | | |

图 2-16　SA 载荷的格式

图中，发起者 Cookie 和响应者 Cookie 是公有信息(如源/目的 IP 地址、UDP 源/目的端口)和一些本地信息的散列值，用以标示本次 ISAKMP 交换，它们可作为一种在场的证据使用；下一载荷表示紧接的载荷类型；接下来是 ISAKMP 的主版本号和从版本号；标志字段主要用于加密同步。

下一载荷表示紧跟在 ISAKMP 头部之后的第一个载荷类型值。目前定义了 13 种载荷，类型值如表 2-1 所示。

尽管 SA 载荷后面一定跟有一个或多个提议载荷(proposal payload)，但下一载荷指向的并不是提议载荷，也不是提议载荷后紧接的变换载荷(transform payload)，而是指向

**表 2-1　ISAKMP 定义的载荷类型**

| 载荷类型 | 值 |
|---|---|
| None | 0 |
| SA 载荷(security association) | 1 |
| 提议载荷(proposal) | 2 |
| 变换载荷(transform) | 3 |
| 密钥交换载荷(key exchange) | 4 |
| 身份载荷(identification) | 5 |
| 证书载荷(certificate) | 6 |
| 证书请求载荷(certificate request) | 7 |
| Hash 载荷(hash) | 8 |
| 签名载荷(signature) | 9 |
| Nonce 载荷(nonce) | 10 |
| 通知载荷(notification) | 11 |
| 删除载荷(delete) | 12 |
| 厂商载荷(vendor) | 13 |
| 保留 | 14～127 |
| 私有用途 | 128～255 |

变换载荷后紧接的载荷。人们习惯将变换载荷看作提议载荷的一部分，而将提议载荷看作 SA 载荷的一部分。如果 SA 载荷已经是最后一个载荷，则下一载荷字段取值为 0。DOI 表示该消息使用哪个 DOI 来解释，DOI 取值为 0 表明该消息使用 ISAKMPDOI 来解释。形式字段用于对提议载荷进行提议。

主模式交换过程描述如下：

① 在消息的第一次交换过程中，主要完成在发起者和响应者之间达成一系列安全参数的共识，并对交换的其余部分拟定规范。它们也会交换每一条消息的 ISAKMP 头中存在的小甜饼(Cookie)——发起者在第一条消息中选择，将其放在 ISAKMP 头的“发起者小甜饼”部分；而响应者在第二条消息中选择，将其放在 ISAKMP 头的“响应者小甜饼”部分。

② 在消息的第二次交换中，通信双方会交换 Diffie-Hellman 公共值(在第一次消息交换时作为保护套件的一部分协商好的组内交换)以及伪随机 Nonce。在这个时候，通信双方可完成它们的 Diffie-Hellman 交换，参与通信的双方会生成 4 种秘密：SKEYID，后续所有的密钥材料都建立在它的基础上；SKEYID_d，用于为 IPSec 衍生出密钥材料；SKEYID_a，用于保障 IKE 消息的数据完整性以及对数据源的身份验证；SKEYID_e，用于对 IKE 消息加密。通信双方将共享 Diffie-Hellman 的共享密钥：g^xy 也会在 SKEYID 的生成中用到。SKEYID 的生成取决于采用的是何种验证机制，而其他所有以 SKEYID 为基础的密钥都以相同的方式衍生出来。对每种验证方法分别计算 SKEYID 值。

对于预共享密钥方式：

$$SKEYID = PRF\ (\text{预共享密钥}, Ni_b \mid Nr_b) \tag{2-1}$$

对于公钥签名方式：

$$SKEYID=PRF(Ni_b|Nr_b,g\hat{}xy) \quad (2\text{-}2)$$

对于公共密钥加密方式：

$$SKEYID=PRF(Hash(Ni_b|Nr_b),CKY_I|CKY_R) \quad (2\text{-}3)$$

其中，|表示连接，PRF(即伪随机函数)通常是协商好的散列函数的一个 HMAC 版本；Ni_b 表示发起方 Nonce 载荷体(即载荷内容)，以下类似。CKY_I 和 CKY_R 表示通信双方的小甜饼(Cookie)。生成了 SKEYID 之后，其他的密钥便可在它的基础上逐次生成。

$$SKEYID_d = PRF(SKEYID,g\hat{}xy \mid CKY_I \mid CKY_R \mid 0) \quad (2\text{-}4)$$

$$SKEYID_a = PRF(SKEYID,SKEYID_d \mid g\hat{}xy \mid CKY_I \mid CKY_R \mid 1) \quad (2\text{-}5)$$

$$SKEYID_e = PRF(SKEYID,SKEYID_a \mid g\hat{}xy \mid CKY_I \mid CKY_R \mid 2) \quad (2\text{-}6)$$

显然，PRF 函数的块长度决定了最终生成的 SKEYID 的长度。假如 PRF 的输出太小，不能作为一个加密密钥来使用，那么 SKEYID_e 必须进行扩展。例如，由 HMAC-MD5 生成的 SKEYID_e 长度是 128b，而 AES 算法可能需要 448b 的密钥，使用下面的扩展方法，取 Ka 的前 448b 为加密密钥。

$$Ka=K1|K2|K3$$

$$K1=PRF(SKEYID_e,0)$$

$$K2=PRF(SKEYID_e,K1)$$

$$K3=PRF(SKEYID_e,K2)$$

③ 在最后一次消息交换过程中，通信双方各自标定自己的身份，并相互交换验证散列摘要。交换的最后两条消息是用 SKEYID_e 进行加密的。为了验证交换中的双方，协议的发起者产生 Hash_I，响应者产生 Hash_R，在采用数字签名认证的主模式和野蛮模式中，发起方和响应方的数字签名 SIG_I 和 SIG_R 是用协商好的数字签名算法分别对 Hash_I 和 Hash_R 签名而产生的。其中：

$$Hash_I=PRF(SKEYID,g\hat{}xi|g\hat{}xr|CKY_I|CKY_R|SAi_b|IDii_b) \quad (2\text{-}7)$$

$$Hash_R=PRF(SKEYID,g\hat{}xr|g\hat{}xi|CKY_R|CKY_I|SAi_b|IDir_b) \quad (2\text{-}8)$$

至此，6 条消息交换完成以后，一个 ISAKMPSA 就成功地建立了。在整个建立过程中，依据双方的安全策略、环境配置，形成了用于保护通信双方后续 IPSecSA 协商的各个安全参数：鉴别方式、加密算法、散列算法、加密/鉴别密钥、Oakley 群、ISAKMPSA 生存期、ISAKMPSA 的标识符等。整个协商过程受到了完整性保护，身份等敏感信息受到了机密性保护。通过 Cookie 的使用，一定程度地防止了协商过程中的拒绝服务攻击。

2) 积极模式

积极模式交换的用途与主模式交换相同以建立一个验证的安全联盟和密钥，随后可用 IKE 为其他安全协议建立安全联盟。主要的差别在于，积极模式只需用到主模式一半的消息。由于对消息的数量进行了限制，积极模式同时也限制了它的协商能力，而且不会提供身份保护。若采用数字签名认证方式，积极模式的交换过程如图 2-17 所示。

在积极模式交换过程中，发起者会提供一个 SA 列表、Diffie-Hellman 公共值、Nonce 以及身份信息。所有这些都是随第一条消息传送的。作为响应者，则需要回应一个选定的 SA、Diffie-Hellman 公共值、Nonce、身份信息以及一个验证载荷(对于预共享密钥以及

图 2-17　积极模式交换过程

加密的 Nonce 验证来说，是一个散列载荷；而对基于签名的验证来说，则是一个签名载荷）。发起者将它的验证载荷作为最后一条消息来传送。

显而易见，积极模式的功能非常有限，在需要进行远程访问的时候，由于发起者的地址不可能被响应者提前知道，而且双方都打算使用预共享密钥验证方法，那么要想建立 IKESA，这便是唯一可行的交换方法。

#### 2. IKE 第二阶段的协商

IKE 第二阶段的协商是通过快速模式来实现的。快速模式的交换必须以第一阶段协商的 ISAKMPSA 来保护，即除了 ISAKMP 报头外，所有的负载都要由 SKEYID_e 加密。

在一个 ISAKMPSA 的保护下，可进行多个快速模式交换。该模式通过交换 3 条消息完成 IPSecSA 的协商。其交换过程如图 2-18 所示。



图 2-18　快速模式交换过程

其中：

Hash1＝PRF(SKEYID_a,M-ID|SA|Ni[|KE][IDci|IDcr])

Hash2＝PRF(SKEYID_a,M-ID|Ni_b|SA|Nr[|KE][IDci|IDcr])

Hash3＝PRF(SKEYID_a,0|M-ID|Ni_b|Nr_b)

IDci,IDcr 表示第二阶段协商双方的身份，从发起者到响应者的第一条消息中包含了在主模式协商中已经看到的负载，当然，这些负载的内容是不一样的；SA 载荷是根据本地策略要求，提议的一种或多种保护数据信息的安全协议（如 ESP 或 AH），并给出其相应的变换（即安全协议的安全参数）；Hash 载荷用于对交换的完整性作出保护，被置于 SA 载荷之前，将首先被检查；Nonce 载荷 Ni 的目的在于给对方一个“在场”的证据，它表明自身确实是通信的对方，而不是冒充者；是否包含可选的 KE 载荷将影响本次 IKE 协商是否能提供 PFS(perfect forward service)服务，PFS 是一种强制对等体双方（如果双方都同意的话）在快速模式交换中产生新的 D-H 秘密属性，这允许使用新的 D-H 密钥生成用于加密数据的加密密钥，以保证更强的安全性。即使攻破第一阶段的密钥交换（即攻破了 SKEYID 等衍生密钥），也只能阅读受该 SA 保护的信息，却不能阅读受 IPSecSA 保护

的信息。可选的身份载荷与第一阶段的身份载荷的作用是不一样的，它不用于鉴别目的，而在于为将建立的IPSecSA协商构造选择符。这个选择符规定了该IPSecSA将保护什么样的通信。

从响应者到发起者的第二条消息用于响应发起者发出的第一条消息。所包含的载荷与第一条消息基本上是一样的，只有个别变化，特别明显的是只有已被接受的提议与变换载荷被返回给发送者。

最后一条消息用来认证响应者的当前活跃性。在这条消息中，发起者需要同时包括Nonce和此次交换的消息ID，并把它们保存在一个验证散列载荷Hash3中。发送给响应者，这样响应者便可知道发起者是否接收到了它在快速模式第一条也是唯一一条消息并正确处理了它，以及避免攻击者发起的有限DoS攻击。注意，所有这3条消息都受到密钥SKEYID_e加密保护。

在第3条消息被响应者接收到以后，IPSec快速模式结束，至此，一个IPSecSA建立完成。整个建立过程受到ISAKMPSA的机密性、完整性保护。而且，通过HDR中的Cookie等字段以及Nonce载荷，使整个建立过程能一定程度地抗重放和拒绝服务攻击。现在双方都同意的IPSecSA可以用来进行安全通信。

### 3. IPSec 协议实施方式

IPSec协议可在终端主机、网关/路由器或两者中同时进行实施和配置。RFC2401中规定了3种标准的IPSec实现方式：

(1) IPSec与操作系统(OS)集成在一起，当作网络层的一部分来实现，负责从IP堆栈中取出数据包，处理后再将其插入，如图2-19(a)所示。

(2) IPSec作为一个“楔子”来实施，插入网络层和数据链路层之间，负责从IP堆栈中取出数据包，处理后再将其插入，如图2-19(b)所示，称为“堆栈中的块(BITS)”。

(3) 在路由器中实现时，IPSec的实现在一个设备中进行，通常是一个外置的专用加密设备直接接入路由器的物理接口，如图2-19(c)所示，称为“线缆中的块(BITW)”。

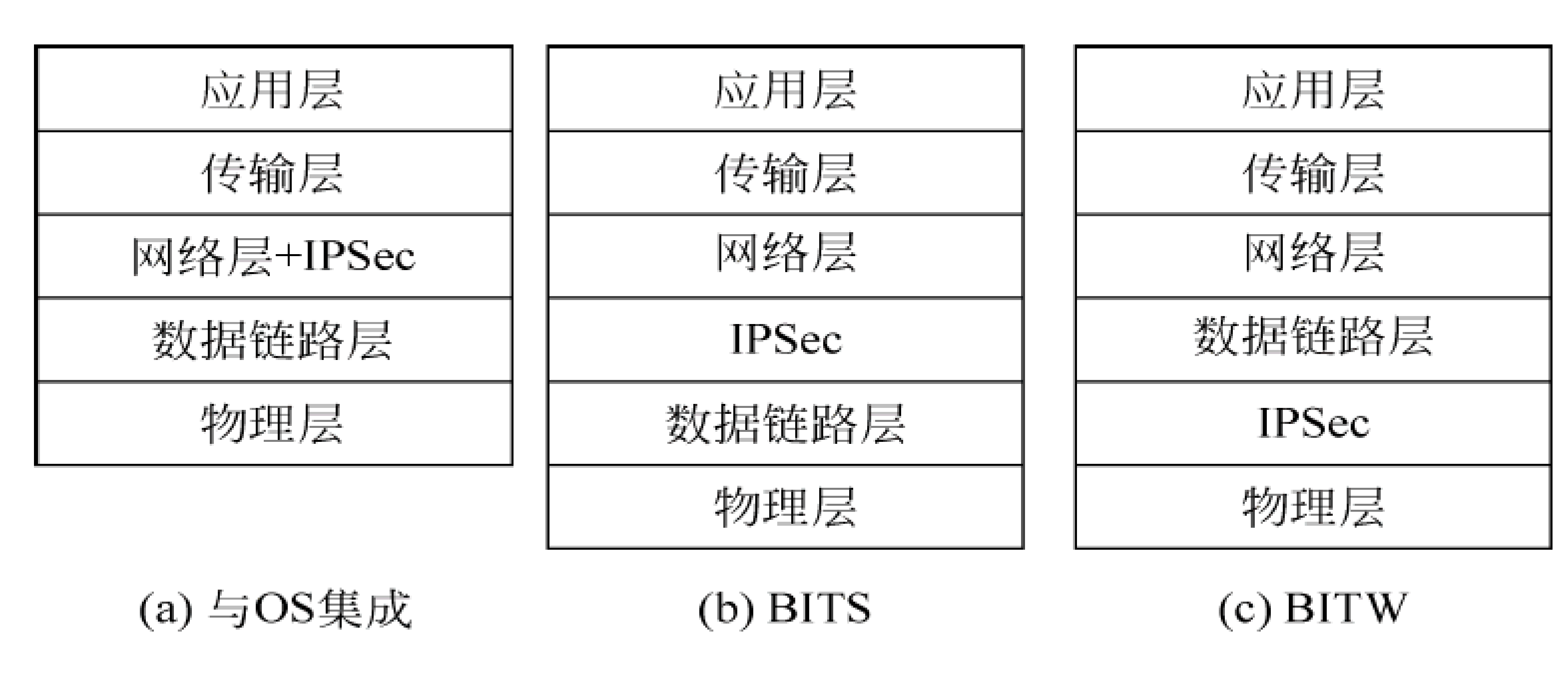



图2-19 IPSec实施方式对协议栈的影响

与操作系统集成实现的IPSec对主机和安全网关都适用，由于IPSec与网络层紧密集成在一起，因此它更有利于诸如分段、PMTU之类的网络服务，这使得实施方案更为有效。另外，由于IPSec实现于操作系统内核，具有内核级别的调度优先级，因此效率很高。

此方式在每个数据流(比如一个 Web 事务处理)的级别提供安全服务更为容易，因为密钥管理、基本 IPSec 协议和网络层可以无缝集成在一起，并且支持所有的 IPSec 模式。

BITS 方式的实现时，IPSec 实现于现存的 IP 协议栈和本地网络驱动程序之间，这种实现不需要得到 IP 协议栈的源代码，对一些旧系统来说比较适合，该方式通常被用于主机上的实现。BITW 式的实现方式在军事和商业系统中经常采用，是一种使用独立的加/解密处理器的实现方式。这种实现方案可以用在主机/网关上，BITW 设备通常是 IP 可寻址的。BITW 设备直接接入主机/网关的物理接口，它一般不运行任何路由算法，只是用来保障数据包的安全。

#### 2.4.6.3　单向 Hash 函数

单向 Hash 函数有很多名字，如压缩函数、缩短函数、消息摘要、指纹、密码校验和、信息完整性检验(DIC)、操作检验码(MDC)。不管怎么称呼，它是现代密码学的中心。单向 Hash 函数是许多协议的另一个结构模块。

Hash 函数长期以来一直在计算机科学中使用，无论从数学上或别的角度看，Hash 函数就是把可变输入长度的字符串转换成固定长度(通常比输入更短)的输出的字符串(叫做 Hash 值)的一种函数。

简单的 Hash 函数就是对输入进行处理，返回由所有输入字节异或组成的一字节。

这里的关键就是采集输入字符串的指纹：Hash 函数生成一个值，利用这个值能够判断出候选的输入是否与真实的输入有相同的值。

因为 Hash 函数是典型的多到一的函数，不能用来确定两个串一定相同，但可用它来得到准确性的合理保证。

单向 Hash 函数是在一个方向上工作的 Hash 函数，从输入的值很容易计算其 Hash 值，但要寻找一个输入使其 Hash 值等于一个特殊值却是很难的。

综上所述，单向 Hash 函数 H(M)将任意长度的消息 M 转换为固定长度 m 的消息：

$$H(M)=h$$

其中 h 的长度恒等于 m。函数 H 具有如下性质：

(1) 给定 M，容易计算 h。

(2) 给定 h，根据 $h=H(M)$很难推出 M。

(3) 给定 M，很难寻找另一个 $M'$，使 $H(M)=H(M')$。

前面提到的简单 Hash 函数不是单向函数：已知一个特殊的字符串，要产生一个字符串使异或结果等于那个值是很容易的事情。但是用单向 Hash 函数就不可能那样做。好的 Hash 函数也是无冲突的：难以找出两个输入值，使它们的 Hash 值相同。

Hash 函数是公开的，对处理过程不保密。单向 Hash 函数的安全性是它的单向性。输入字符串的单个比特的改变，平均而言，将引起 Hash 值中一半的比特改变。已知一个 Hash 值，要找到其 Hash 输入的值，使它的 Hash 值等于已知的 Hash 值在计算上是不可行的。

单向 Hash 函数可以看作是构成指纹文件的一种方法。如果 A 想验证某人持有一特定的文件(同时 A 也持有该文件)，但又不想让对方将文件传给自己，就可以要求对方将该文件的单向 Hash 值传送过来。如果对方传送的 Hash 值是正确的，那么可以肯定地说

对方持有那份文件。

这是在金融交易中的特殊使用,如果不希望在网络的一些地方把提取 100 美元变成提取 1000 美元。一般情况下,应使用不带密钥的单向 Hash 函数,以便任何人都能验证 Hash 值。如果只想接收者才能验证 Hash 值,那么用消息认证码就可以了。

#### 2.4.6.4 消息认证码

消息认证码(message authentilation codes, MAC)也叫数据认证码(date authentilation codes,DAC),是带有秘密密钥的单向 Hash 函数。Hash 值是输入密钥的函数。这在理论上与 Hash 函数一样,但是只有拥有密钥的人才能验证 Hash 值。可以用 Hash 函数或分组加密算法产生 MAC,也有专用于 MAC 的算法。

## 2.5 本章重点和难点

本章的重点是 Internet 协议,以及 Internet 的协议为什么不安全,目前在 Internet 中实施安全保护的层次,IPSec 的基本概念。

第 2.1 节 Internet 协议部分的重点是了解协议的基本构架和定址与路由部分;第 2.2 节是本章的重点;第 2.3 节的重点是了解在各层实施安全保护的优缺点;第 2.4 节的重点是了解 IPSec 的体系结构,理解密钥交换协议的过程。

本章难点是如何通过具体事例获得关于安全协议的感性知识。

## 习题与思考题

1. 试述网络协议体系的构成。
2. 简述 IPv4、IPv6 的定址方法。
3. 现存 TCP/IP 协议的安全缺陷有哪些?
4. 试述安全协议的基本要求。
5. 试述 L2TP 与 IPSec 的联系。
6. 通过对各层安全协议的了解,举例说明密码算法在协议的安全性上所起的重要作用。
7. Internet 安全关联和密钥管理协议 IKE 的目的和作用是什么?
8. 如何区别安全关联?
9. IP 层的安全体系结构(IPSec)包含哪些安全服务,其实现的基础是什么?
10. 说明 IP 认证头 AH 的作用和格式。
11. 说明封装安全载荷协议 ESP 的作用和格式。
12. IP 层的数据交换模式可以分为传输模式和隧道模式,请分别说明其过程和特点。
13. 在 AH 和 ESP 中,是怎样实现重放攻击的?
14. AH 和 ESP 的传输模式和隧道模式的主要区别是什么?
15. 当两个传输模式 SA 进行捆绑,以允许在相同端数据流中有 AH 和 ESP 协议,应在执行 AH 前执行 ESP,为什么?

# 第 3 章 安全协议的密码学基础

安全协议中需要使用各种密码算法。本章对此作一介绍。

本章共有 6 个小节，第 3.1 节介绍安全协议与密码学的关系；第 3.2 节介绍密码算法；第 3.3 节介绍利用密码算法建立安全通信信道；第 3.4 节介绍不适用密码算法的安全协议；第 3.5 节介绍 Hash 算法的使用；第 3.6 节是本章重点和难点分析。

安全协议，有时也称作密码协议，是以密码学为基础的消息交换协议，其目的是在网络环境中提供各种安全服务。安全目标是多种多样的。例如，认证协议的目标是认证参加协议的主体的身份。此外，许多认证协议还有一个附加的目标，即在主体之间安全地分配密钥或其他各种秘密。

## 3.1 安全协议与密码学的关系

安全性是指数据安全性、通信安全性、信息安全性等，就像一条链子，整个系统的安全性由最脆弱的连接的安全性所决定。因此链子上每一处都必须安全，才能保证整个链子的安全(加密算法、协议、密钥管理、使用规范等)，任何一环出了问题，都会危及整个链子。如果算法很好，但是协议存在漏洞，那么攻击通过该协议漏洞就可能破坏系统。

所以，密码学仅是安全性的一部分，甚至是很小的一部分。它仅在数学上使一个系统安全，这与实际使系统安全是不同的。

此外，对计算机密码学，世界上绝大部分密码学并没有用来保护军事机密，而用于诸如银行卡、付费电视、道路收费、办公大楼及计算机访问令牌、抽彩设备、预付款电子计量器等。在这些应用中，密码的作用就是使卑鄙的犯罪更困难，对那些高额聘请有才能的大量密码分析者和满屋子计算机的攻击者并不适用。这些应用大部分使用了性能差的密码算法，但是被攻破的原因与密码分析没有多大关系，而与受欺骗的雇员、聪明的敲诈行为、愚蠢的实现、频繁地说漏嘴、随便的举止等有关。甚至 NSA 也承认，在它关注领域的大多数安全失败是由工作运作错误引起的，而不是算法或协议上的失败。在这些场合，密码算法再好也没有什么用处，完全有可能绕过它。

所以，密码学(cryptography)与网络安全(network security)之间有差别，也有联系。密码学是一种以秘密的方式编码信息，使只有知道编码秘密的接收者才可以解密被编码的信息的方法。

网络安全得益于密码学的应用，如基于网络的用户登录协议。不过，它也并不是必须依赖于密码学的应用，例如，Unix 系统中对文件的访问控制，就没有使用密码。反过来，密码学在网络中的正常应用也依赖于系统的安全。密码算法常常用软件或硬件实现，它

们能否正常运作关键取决于是否有一个安全的系统。比如,如果系统缺乏访问控制的安全性,攻击者可以修改密码系统的软件算法,造成系统失密。可见,安全性的缺乏会直接影响密码技术的效用。

网络安全经历了二十多年的发展,已经成为一个跨多门学科的综合性科学,它包括通信技术、网络技术、计算机软硬件设计技术、密码学、网络安全与计算机安全技术等。

目前的网络安全从理论上讲是建立在密码学以及网络安全协议的基础上的。密码学是网络安全的核心,利用密码技术对信息进行加密传输、加密存储、数据完整性认证、用户身份认证等,比传统意义上简单的存取控制和授权等技术更可靠。加密算法是一些公式和法则,它规定了明文和密文之间的变换方法。由于加密算法的公开化和解密技术的发展,加上发达国家对关键加密算法的出口限制,各国正不断致力于开发和设计新的加密算法和加密机制。

安全协议方面,众多标准化组织制定了许多标准和草案,尤其是以 RFC 文档出现的协议标准更是网络安全设备的基础。因此,要不断发展和开发满足新的需求的安全协议。

从技术上,网络安全取决于两个方面:网络设备的硬件和软件,网络安全由网络设备的软件和硬件互相配合来实现。但是,由于网络安全对网络上的信息提供一种增值服务,人们往往发现软件的处理速度成为网络的“瓶颈”,因此,将网络安全的密码算法和安全协议用硬件实现,实现线速的安全处理是网络安全的一个主要方向。

另外,在安全技术不断发展的同时,全面加强安全技术的应用也是网络安全发展的一个重要内容。因为即使有了网络安全的理论基础,没有广泛地将它应用于网络中,那么谈再多的网络安全也是无用的。同时,网络安全不仅仅是密码算法、防病毒、入侵监测、防火墙、身份认证、加密等产品的简单堆砌,而是包括从系统到应用、从设备到服务的比较完整的、体系性的安全系列产品的有机结合。

除了提供对消息的保护外,密码学在网络安全中通常还有其他作用。

- 认证:消息的接收者应该能够确认消息的来源,入侵者不可能伪装成他人。
- 完整性:消息的接收者应该能够验证在传送过程中消息没有被修改,入侵者不可能用假消息代替合法消息。
- 抗抵赖:发送者事后不可能否认他发送的消息。

这些功能需要通过计算机完成。某人是否就是他说的人;某人的身份证明文件(身份证、学历或者护照)是否有效;声称从某人那里来的文件是否确实从那个人那里来的,这些事情都是通过认证、完整性和抗抵赖来实现的,就像面对面交流一样。

## 3.2 密码算法

目前,安全协议采用密码算法来保护秘密信息。密码算法是用于加密和解密的数学函数。通常,有两个相关的函数:一个用作加密,另一个用作解密。

如果算法的保密性是基于保持算法的秘密,这种算法称为受限制的算法。受限制的算法具有历史意义,但按现在的标准,它们的保密性已远远不够。大的或经常变换的用户组织不能使用它们,因为每当有一个用户离开这个组织,其他的用户就必须改用另外的算

法。如果有人无意暴露了这个秘密，所有人都必须改变其算法。

此外，受限制的密码算法不可能进行质量控制或标准化。每个用户组织必须有他们自己的唯一算法。这样的组织不可能采用流行的硬件或软件产品。但窃听者却可以买到这些流行产品并学习算法，于是用户不得不自己编写算法并予以实现，如果这个组织中没有好的密码学家，那么就无法知道他们是否拥有安全的算法。

尽管有这些主要缺陷，受限制的算法对低密级的应用来说还是很流行的，用户或者没有认识到或者不在乎他们系统中内在的问题。

现代密码学用密钥解决了这个问题，密钥用 K 表示。K 可以是很多种数值里的任意值。密钥 K 的可能值的范围叫做密钥空间。加密和解密运算都使用这个密钥（即运算都依赖于密钥，并用 K 作为下标表示），这样，加/解密函数变成：

$$E_K(M) = C \tag{3-1}$$

$$D_K(C) = M \tag{3-2}$$

这些函数具有下面的特性，如图 3-1 所示：

$$D_K(E_K(M)) = M \tag{3-3}$$

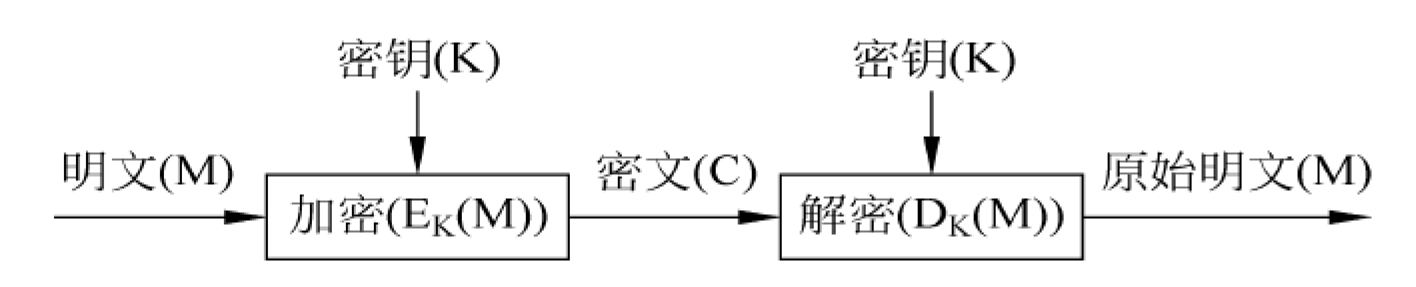



图 3-1　使用一个密钥的加/解密

有些算法使用不同的加密密钥和解密密钥，也就是说加密密钥 $K_1$ 与相应的解密密钥 $K_2$ 不同，如图 3-2 所示。

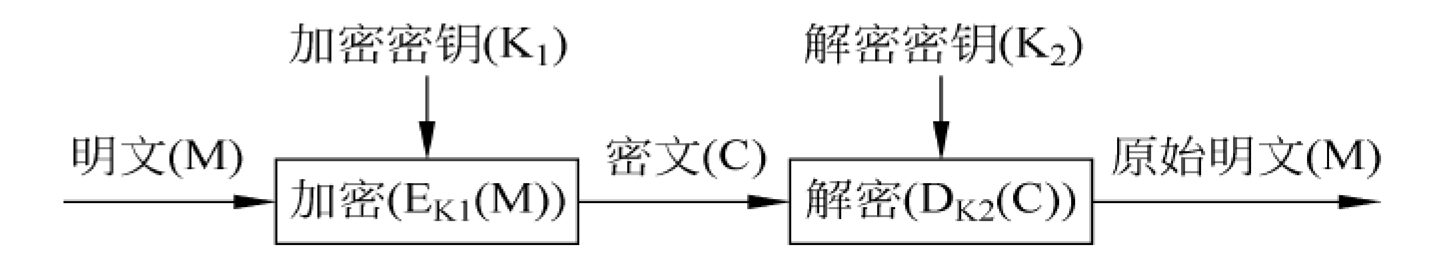



图 3-2　使用两个密钥的加/解密

$$E_{K1}(M) = C \tag{3-4}$$

$$D_{K2}(C) = M \tag{3-5}$$

$$D_{K2}(E_{K1}(M)) = M \tag{3-6}$$

所有这些算法的安全性都基于密钥的安全性，而不是基于算法的细节的安全性。这就意味着算法可以公开，也可以被分析，可以大量生产使用算法的产品，即使偷听者知道使用的算法也没有关系；如果不知道使用的具体密钥，就不可能阅读加密消息。

综上所述，密码系统是由解密、解密算法以及所有可能的明文、密文和密钥组成的。

基于密钥的算法通常有两类：对称算法和非对称（公开密钥）算法。还有一类算法，有时称为 Hash 函数，用于对数据进行完整性验证。

### 3.2.1　对称密码算法

对称密码算法又叫传统密码算法，就是加密密钥能够从解密密钥中推算出来，反过来

也成立。在大多数对称算法中，加密解密密钥是相同的。这些算法也叫秘密密钥算法或单密钥算法，它要求发送者和接收者在安全通信之前，商定一个密钥。

对称算法的安全性依赖于密钥，泄露密钥就意味着任何人都能对消息进行加密解密。只要通信需要保密，密钥就必须保密。对称算法的加密和解密表示为：

$$E_K(M) = C \tag{3-7}$$

$$D_K(C) = M \tag{3-8}$$

式中，M 表示消息；K 表示密匙；C 表示加密后的消息；E 表示加密函数；D 表示解密函数。

对称算法可分为两类。一类只对明文中的单个位（有时对字节）进行运算的算法称为序列算法或序列密码；另一类算法是对明文的一组位进行运算，这些位组被称为分组，相应的算法称为分组算法或分组密码。现代计算机密码算法的典型分组长度（例如 DES）为 64 位，这个长度大到难以分析破译，但又小到方便作用。目前，比较高级的加密应用是使用 AES 变长加密，包括 128 位、192 位、256 位 3 种方式。

这种算法具有如下的特性：

$$D_K(E_K(M)) = M \tag{3-9}$$

常用的对称密码术的加密方案有 5 个组成部分（如图 3-3 所示）。

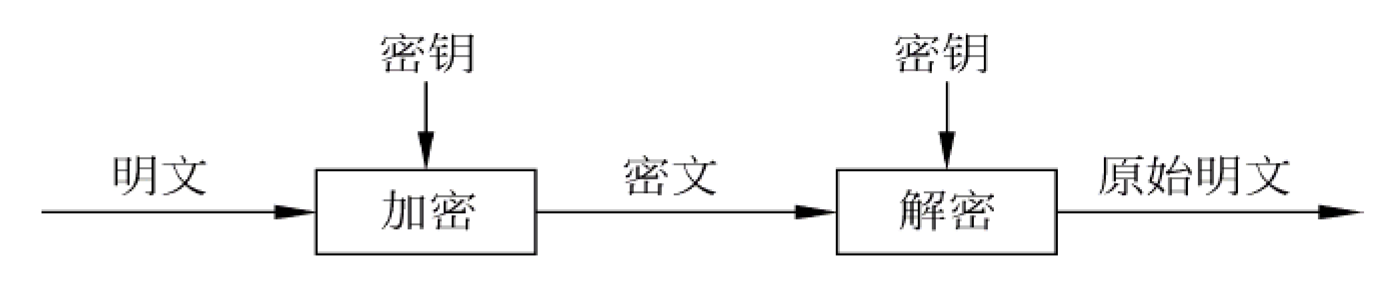



图 3-3 采用对称密码的加密

其中：

明文——原始信息。

加密算法——以密钥为参数，对明文进行多种置换和转换的规则和步骤，变换结果为密文。

密钥——加密与解密算法的参数，直接影响对明文进行变换的结果。

密文——对明文进行变换的结果。

解密算法——加密算法的逆变换，以密文为输入、密钥为参数，变换结果为明文。

对称密码术的优点在于效率高（加/解密速度能达到数十兆字节每秒或更多），算法简单，系统开销小，适合加密大量数据。

尽管对称密码术有很好的特性，但也存在着明显缺陷，包括：

(1) 进行安全通信前需要以安全方式进行密钥交换。这一步骤，在某种情况下是可行的，但在某些情况下会非常困难，甚至无法实现。

(2) 规模复杂。举例来说，A 与 B 两人之间的密钥必须不同于 A 和 C 两人之间的密钥，否则给 B 的消息的安全性就会受到威胁。在有 1000 个用户的团体中，A 需要保持至少 999 个密钥（更确切地说是 1000 个，需要留一个密钥给自己加密数据）。

对于该团体中的其他用户，此种情况同样存在。从而这个团体一共需要将近 50 万个不同的密钥！依此类推，n 个用户的团体需要 $n^2/2$ 个不同的密钥。

通过应用基于对称密码的密钥中心结构，上述问题可以有所缓解。在这种结构中，团体中的任何一个用户与中心服务器(通常称作密钥分配中心)共享一个密钥。因而，需要存储的密钥数量基本上和团体的人数差不多，而且中心服务器也可以为以前互相不认识的用户充当“介绍人”。但是，这个与安全密切相关的中心服务器必须随时都是在线的，因为只要服务器掉线，用户间的通信将不可能进行。这就意味着中心服务器是整个通信成败的关键和受攻击的焦点，也意味着它还是一个庞大组织通信服务的“瓶颈”。

## 3.2.2 非对称密码算法

非对称密码术也被称作公钥密码术，其思想是由 W. Diffie 和 Hellman 在 1976 年提出的。不同于以往的加密技术，非对称密码术是建立在数学函数基础上的，而不是建立在位方式的操作上的。更重要的是，与只使用单一密钥的传统加密技术相比，它在加/解密时，分别使用两个不同的密钥：一个可对外界公开，称为“公钥”；一个只有所有者知道，称为“私钥”。公钥和私钥之间紧密联系，用公钥加密的信息只能用相应的私钥解密，反之亦然。同时，要想由一个密钥推知另一个密钥，在计算上是不可能的。

其加密算法有如下特性：

$$E_{K1}(M) = C \tag{3-10}$$

$$D_{K2}(C) = M \tag{3-11}$$

$$D_{K2}(E_{K1}(M)) = M \tag{3-12}$$

其中，M 表示消息；$K_1$ 表示公匙；$K_2$ 表示私匙；C 是加密后的消息；E 是加密函数；D 为解密函数。

如果规定公开的密钥用于加密数据，而私钥用于数字签名，则在采用公钥体制下 A 向 B 传输数据的过程如图 3-4 所示。

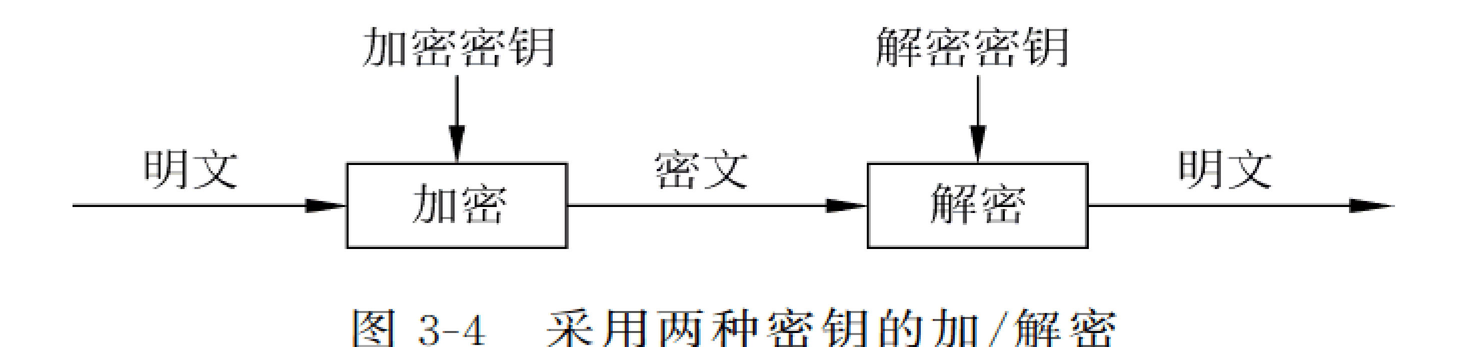



图 3-4　采用两种密钥的加/解密

加/解密的过程为：

(1) A 查找 B 的公钥。因为公钥的公开不会影响到通信的保密性，B 可以将自己的公钥公布在公共数据库，由其他人取用，或以普通电子邮件等方式通过非安全信道发送给 A。

(2) A 采用公钥加密算法以 B 的公钥作为加密密钥对原始信息进行加密。

(3) A 通过非安全信道将密文发送给 B。

(4) B 收到密文后，使用自己持有的私钥对其解密，还原出明文。

从以上的介绍中可以看出，与对称密码技术相比较，利用非对称密码技术进行安全通信，有以下优点：

(1) 通信双方事先不需要通过保密信道交换密钥。

(2) 密钥持有量大大减少。在 n 个用户的团体中进行通信，每一用户只需要持有自

己的私钥，而公钥可放置在公共数据库上，供其他用户取用。这样，整个团体仅需 n 对密钥，就可以满足相互之间进行安全通信的需求。

(3) 非对称密码技术还提供了对称密码技术无法或很难提供的服务：如与 Hash 函数联合运用可组成数字签名，可证明的安全伪随机数发生器的构造，零知识证明等。使用非对称密码技术的主要缺点是：加/解密速度慢、耗用资源大。一般来说，实用的加/解密方案都综合运用了对称密码技术和非对称密码技术。

### 3.2.3 Hash 算法

Hash 算法也称 Hash 函数，Hash 函数有多个名称，最常用的是哈希函数、散列函数、消息摘要函数、单项函数、压缩函数、缩短函数等。本书中统一称为 Hash 算法。后面将要用到的有关概念还有如下两个。

(1) Hash 值：指对某个输入利用 Hash 算法进行运算之后得到的值，一般是某个固定长度的值。

(2) Hash 运算：指对某个输入利用 Hash 算法进行运算。

在计算机安全系统中 Hash 算法广泛用于数字签名、消息的完整性认证、消息的起源认证等。

在信息安全技术中，经常需要验证消息的完整性，通常利用 Hash 函数来完成这种服务。简单地说，Hash 变换是一种有损压缩，就是一种将任意长度的消息压缩到某一固定长度的函数。也就是说，Hash 算法对不同长度的输入消息，产生固定长度的输出。这个固定长度的输出称为原输入消息的“散列”或“消息摘要”(Message-digest)。

一个安全的 Hash 算法 H 必须具有以下属性：

(1) H 能够应用到大小不一的数据上。

(2) H 能够生成大小固定的输出。

(3) 对于任意给定的 x，H(x)的计算相对简单。

(4) 对于任意给定的代码 h，要发现满足 H(x)=h 的 x 在计算上是不可行的。

(5) 对于任意给定的块 x，要发现满足 H(y)=H(x)而 y=x 在计算上是不可行的。

(6) 要发现满足 H(x)=H(y)的(x,y)对在计算上是不可行的。

### 3.2.4 一次一密乱码本

有一种理想的加密方案，称为一次一密乱码本。一般地，一次一密乱码本本质上是一个大的不重复的真随机密钥字母集，这个密钥字母集可以被写在几张纸上，并一起粘成一个乱码本。它最初的应用形式是用于电传打字机。发送方用乱码本中的每一密钥字母准确地加密一个明文字符。加密是明文字符和一次一密乱码本密钥字符的模 26 加法。

每个密钥仅对一个消息使用一次。发送方对所发的消息加密，然后销毁乱码本中用过的一页或用过的纸带部分。接收方有一个同样的乱码本，并依次使用乱码本上的每个密钥去解密密文的每个字符。接收方在解密消息后销毁乱码本中用过的一页或用过的纸带部分。新的消息则用乱码本的新的密钥加密。

例如，如果消息是：

ONETIMEPAD，

而取自乱码本的密钥序列是

TBFRGFARFM，

那么，密文就是

IPKLPSFHGQ，

因为

$$(O+T)\bmod 26=I$$
$$(N+B)\bmod 26=P$$
$$(E+F)\bmod 26=K$$

等等。

如果窃听者不能得到用来加密消息的一次一密乱码本，这个方案是完全保密的。给出的密文消息相当于同样长度的任何可能的明文消息。

由于每一密钥序列都是等概的（注意：密钥是以随机方式产生的），敌方没有任何信息用来对密文进行密码分析，密钥序列也可能是：

POYYAEAAZX

解密出来是：

SALMONEGGS

或密钥序列为：

BXFGBMTMXM

解密出来的明文为：

GREENFLUID

值得重申的是：由于明文消息是等概的，所以密码分析者没有办法确定哪一明文消息是正确的。随机密钥序列异或一非随机的明文消息产生一完全随机的密文消息。再大的计算能力也无能为力。

值得注意的是：密钥字母必须是随机产生的。对这种方案的攻击将是针对用来产生密钥序列的那种方法。使用伪随机数发生器是不值得考虑的，它们通常具有非随机性。如果采用真随机源（这比第一次出现难得多），它就是安全的。

另一个重要的事情是密钥序列不能重复使用，即使用多兆字节的乱码本，如果密码分析者有多个密钥重叠的密文，也能够重构明文。他把每排密文移来移去，并计算每个位置的适配量。如果排列正确，则适配的比例会突然升高（准确的百分比与明文的语种有关）。从这一点来说，密码分析是容易的，它类似于重合指数法，只不过用两个“周期”作比较。所以千万别重复使用密钥序列。

一次一密乱码本的想法很容易推广到二进制数据的加密，只需由二进制数字组成的一次一密乱码本代替由字母组成的一次一密乱码，用异或代替一次一密乱码本的明文字符加法即可。为了解密，用同样的一次一密乱码本对密文异或，其他保持不变，保密性也很完善。

这听起来简单，但先要解决几个问题。因为密钥比特必须是随机的，并且绝不能重复使用，密钥序列的长度要等于消息的长度。一次一密乱码本对短信息是可行的，但它绝不

可能在宽带或大流量的通信信道上工作。虽然能在一张 CD-ROM 中存储 650MB 的随机二进制数,但还有一些问题要注意:首先,需要准确地复制两份随机数比特,CD-ROM 对大量的数据来说是经济的;其次,需要能够销毁已经使用过的比特,而 CD-ROM 没有抹除设备,除非物理毁坏整张盘。数字磁带对这种东西来说是更好的媒体。

即使解决了密钥的分配和存储问题,还需确信发送方和接收方是完全同步的。如果接收方有一比特的偏移(或者一些比特在传送过程中丢失了),消息就会变得乱七八糟。另一方面,如果某些比特在传送中被改变了(没有增减任何比特,更像由于随机噪声引起的),那些改变了的比特就不能正确地解密。再者,一次一密乱码本不提供鉴别。

一次一密乱码本在今天的应用场合主要用于高度机密的低带宽信道。据说,美国和前苏联之间的热线电话就是用一次一密乱码本加密的。许多前苏联间谍传递的消息也是用一次一密乱码本加密的。到今天这些消息仍是保密的,并将一直保密下去。不管超级计算机工作多久,也不管半个世纪中有多少人、用什么样的方法和技术、具有多大的计算能力,他们都不可能阅读前苏联间谍用一次一密乱码本加密的消息,除非得到加密消息的一次一密乱码本。

## 3.3 利用密码算法建立安全通信信道

### 3.3.1 对称密码技术

相互不能见面的通信双方怎样安全地通信呢?对通信的内容进行加密可以实现保密通信。但是完整的协议描述很复杂,下面先来看看 A 发送加密的信息给 B 的过程:

(1) A 和 B 协商使用同一密码系统。

(2) A 和 B 协商使用同一密钥。

(3) A 用加密算法和选取的密钥加密其明文信息,得到密文信息。

(4) A 发送密文信息给 B。

(5) B 用同样的算法和密钥解密密文,然后得到明文信息。

位于 A 和 B 之间的窃听者 E 在监听这个协议。如果听到的是在步骤(4)中发送的密文,E 必须设法分析密文,这是唯密文的被动攻击法;有很多算法能够阻止 E,使其不可能得到问题的解答。

尽管如此,但 E 也在窃听步骤(1)和步骤(2),这样就知道了算法和密钥,E 就和 B 知道的一样多。当步骤(4)中的信息通过信道传送过来时,E 所做的全部工作就是解密密文信息。

好的密码系统的全部安全性只与密钥有关,与算法没有任何关系。因此,密钥管理在密码学中极为重要。

有了对称算法,A 和 B 能够公开地实现步骤(1),但必须秘密地完成步骤(2)。在协议执行前、执行过程中和执行后,只要信息必须保持秘密,密钥就必须保持秘密,否则,信息就将不再是秘密的了。公开密钥密码学用另一种方法解决了这个问题,将在下一节中讨论。

主动攻击者 M 则在做其他事情，例如，企图破坏在步骤(4)中使用的通信信道，使 A 和 B 根本不可能通信。也可以截取 A 的信息并用自己的信息替代它。如果 M 也知道密钥(通过截取步骤(2)的通信或者破译密码系统)，就可能加密自己的信息，然后发送给 B，用来代替截取的信息。B 没有办法知道接收到的信息不是来自 A。如果 M 不知道密钥，所产生的代替信息被解密出来是无意义的，B 就会认为从 A 那里来的信息是网络或者是 A 有严重问题。

此外，A 是否也可以破坏这个协议？显然，A 可以把密钥的副本给 E。这样 E 就可以读 B 所发的信息，但 B 却不知道。虽然问题很严重，但这并不是协议的问题。在协议过程的任何一点都不可能阻止 A 把明文的副本交给 E。当然，类似地，B 也可能做 A 所做的事。所以协议需要假定 A 和 B 是互相信任的。

总之，对称密码算法存在下面的密钥问题。

分配：密钥必须秘密地分配，密钥比任何加密的信息更有价值，因为知道了密钥意味着知道了所有信息。对于遍及世界的加密系统，这是令人难以忍受的任务，需要经常派信使将密钥传递到目的地。

泄露：如果密钥被损害(被偷窃，猜出来，被逼迫交出来，受贿等)，那么 E 就可以用该密钥去解密所有传送的信息，也可以假装是几方中的某一方，产生虚假信息去欺骗另一方。

数量：假设网络中每对用户使用不同的密钥，那么密钥总数随着用户数的增加迅速增多。N 个用户的网络需要 n(n－1)/2 个密钥。例如，10 个用户互相通信需要 45 个不同的密钥，100 个用户需要 4950 个不同的密钥等。虽然这个问题可以通过将用户数量控制在较小数目来减轻，但这又限制了应用。

### 3.3.2　公开密钥密码技术

如果把对称算法看成保险柜，密钥就是保险柜的号码组合。知道号码组合的人能够打开保险柜，放入文件，再关闭它。持有号码组合的其他人可以打开保险柜，取出文件来，而不知道保险柜号码组合的人就必须去摸索打开保险柜的方法。

1976 年，Whitfield Diffie 和 Martin Hellman 改变了密码学的这种范例(虽然 NSA 宣称早在 1966 年就有这种概念的知识，但没有提供证据)。他们提出了公开密钥密码学。他们使用两个不同的密钥：一个是公开的，另一个是秘密的。持有公钥的任何人都可加密信息，但却不能解密。只有持有私钥的人才能解密。就好像有人把密码保险柜变成一个信箱，把邮件投进邮箱相当于用公开密钥加密，任何人都可以做，只要打开窗口，把它投进去。取出邮件相当于用私钥解密。一般情况下，打开它是很难的，需要焊接机和火把。然而，如果拥有私钥(开信箱的钥匙)，就很容易从邮箱中取出邮件。

从数学上来说，这个过程是基于前面讨论过的单向陷门函数。加密使用公开密钥，任何人都能加密信息。解密做得非常困难，以至于如果不知道这个秘密，即使用 Cray 计算机和几百万年的时间都不能解开这个信息。这个秘密或陷门就是私钥。持有这个秘密，解密就和加密一样容易。

下面描述 A 怎样使用公开密钥密码发送信息给 B(第一个协议)：

(1) A 和 B 选用一个公开密钥密码系统。

(2) B 将自己的公钥传送给 A。

(3) A 用 B 的公钥加密自己的信息，然后传送给 B。

(4) B 用自己的私钥解密 A 的信息。

**注意**：公钥密码是怎样解决对称密码系统的密钥管理问题的。在对称密码系统中，A 和 B 不得不选取同一密钥。A 能够随机选取一个，但她不得不把选取的密钥传给 B。她可能事先交给 B，但需要有先见之明。她也可以通过秘密信使把密钥送给 B，但太费时间。采用公钥密码，就很容易了，不用事先安排，A 就能把信息安全地发送给 B。一直都在窃听整个交换过程的 E，有 B 的公钥和用公钥加密的信息，但却不能恢复 B 的私钥或者传送的信息。

网络中的用户约定一公钥密码系统，每一用户有自己的公钥和私钥，并且公钥在某些地方的数据库中都是公开的，现在这个协议就更容易了(第二个协议)：

(1) A 从数据库中得到 B 的公钥。

(2) A 用 B 的公钥加密信息，然后送给 B。

(3) B 用自己的私钥解密 A 发送的信息。

第一个协议中，在 A 给 B 发送信息前，B 必须将自己的公钥传送给 A，第二个协议更像传统的邮件方式，直到 B 想读自己的信息时，才与协议有牵连。

### 3.3.3 混合密码系统

历史上，在讨论把 DES 算法作为标准建议的同时，公布了第一个公开密钥算法。值得一提的是，现在很多高级应用中，DES 已逐渐被 AES 所取代。

在现实世界中，公开密钥算法不会代替对称算法。公开密钥算法不用来加密消息，而用来加密密钥。这样做有两个理由：

(1) 公钥算法比对称算法慢，对称算法比公钥算法快 1000 倍。虽然，计算机变得越来越快，在 15 年后计算机运行公开密钥密码算法的速度比得上现在计算机运行对称密码的速度。但是，带宽需求也在增加，总有比公开密钥密码处理更快的加密数据要求。

(2) 公开密钥密码系统对选择明文攻击是脆弱的。如果 C＝E(P)，当 P 是 N 个可能明文集中的一个明文，那么密码分析者只需要加密所有 N 个可能的明文，并能与 C 比较结果(注意：加密密钥是公开的)。用这种方法，不可能恢复解密密钥，但能够确定 P。

如果持有几个可能加了密的明文消息，那么采用选择明文攻击可能特别有效。例如，如果 P 是比 100 万美元少的某个美元值，密码分析者尝试所有 100 万个可能的美元值，即使 P 不很明确，这种攻击也是非常有效的。仅仅知道密文与某个特殊的明文不相符，就可能是有用的信息。对称密码系统不易受这种攻击，因为密码分析者不可能用未知的密钥来完成加密的尝试。

在大多数实际的实现中，公开密钥密码用来确保安全和分发会话密钥。这些会话密钥用在对称算法中，对通信消息进行保密。有时称这种系统为混合密码系统。

(1) B 将自己的公开密钥 $E_B$ 发给 A：$E_B \to A$。

(2) A 产生随机会话密钥 K，用 B 的公开密钥加密 K，并把加密后的密钥 $E_B(K)$ 送给

B：$E_B(K)\rightarrow B$。

(3) B 用自己的私钥解密 A 的消息，恢复出会话密钥：

$$D_B(E_B(K))=K$$

(4) 两人用同一会话密钥 K 对他们的通信信息进行加密。

把公开密钥密码用于密钥分配解决了很重要的密钥管理问题。对对称密码而言，数据加密密钥直到使用时才起作用。如果 E 得到了密钥，那么她就能够解密用这个密钥加密的消息。在前面的协议中，当需要对通信加密时，才产生会话密钥，不再需要时就销毁，这极大地减少了会话密钥遭到损害的风险。当然，私钥面对泄露是脆弱的，但风险较小，因为只有每次对通信的会话密钥加密时才用它。这将在下一节中进一步讨论。

## 3.4　不使用密码算法的安全协议的例子

历史上，Ralph Merkle 设计了第一个公开密钥密码。

Merkle 的技术基于：发送者和接收者解决难题比窃听者更容易。下面就是根据这一思路提出的，A 不用首先和 B 交换密钥就把加密消息发给 B 的过程：

(1) B 产生 $2^{20}$ 或大约 100 万个这种形式的消息："这是难题数 x，这是秘密密钥数 y"，其中 x 是随机数，y 是随机的秘密密钥。每个消息的 x 和 y 都是不相同的。采用对称算法，用不同的 20 比特密钥对每个消息加密，并都发给 A。

(2) A 随机选择一个消息，通过穷举攻击恢复明文。这个工作量是很大的，但并不是不可能的。

(3) A 用恢复出来的密钥 y 和一条需要传递的用对称算法加密秘密消息，并把它和密钥 y 相应的 x 一起发给 B。

(4) B 知道自己用哪个秘密密钥 y 对消息 x 加密的，这样就能解密消息。

E 要破译这个系统，就必须做比 A 和 B 多得多的工作。为了恢复步骤(3)的消息，E 必须完成对 B 在步骤(1)中的所有 $2^{20}$ 消息的穷举攻击。这个攻击的复杂性是 $2^{40}$。x 的值不会对 E 有什么帮助，在步骤(1)中它们是随机指定的。一般情况下，E 花费的努力大约是 A 花费的努力的平方。

按照密码学的标准，$n\sim n^2$ 没有什么优势，但在某些情况下，这可能足够复杂。如果 A 和 B 每秒可试 1 万个，这将使他们每个人要花 1 分钟去完成各自的步骤，再花另外 1 分钟在 1.544Mb/s 链路上完成从 A 到 B 的通信难题。如果 E 有同样的计算设备，破译这个系统将花费她大约 1 年的时间，其他算法甚至更难破译。

## 3.5　Hash 算法的使用——数字签名

长期以来，文件上的手写签名用于作者身份的证明，或者同意文件的内容的证明。手写签名使人重视的原因是：

(1) 签名是可信的。签名使文件的接收者相信签名者是慎重地在文件上签字的。

(2) 签名不可伪造。签名证明是签字者而不是其他人的签字。

(3) 签名不可重用。签名是文件的一部分,不法之徒不可能将签名移到不同的文件上。

(4) 签名的文件是不可改变的。在文件签名后,文件不能改变。

(5) 签名是不可抵赖的。签名和文件是物理的东西。签名者事后不能声称他没有签过名。

但是在现实生活中,关于签名的上述条件是难以达到的。实际上,签名可以伪造,可以从一份文件盗用移到另一份文件中;文件在签名后还可以篡改等。现实中之所以相信上述说法,因为欺骗是困难的,并且还要冒被发现的危险。

另一方面,针对计算机中的文件,要达到上述信任,问题就变得十分困难。因为,首先计算机文件易于复制,即使某人的签名难以伪造(例如,手写签名的图形),从一个文件到另一个文件剪裁和粘贴有效的签名都是很容易的;其次在签名后,也容易修改文件,并且不会留下任何修改的痕迹。

所以,为了对计算机文件进行有效签名,需要利用加/解密知识才能产生所谓的"数字签名",以确保计算机上文件的有效性。数字签名是目前电子商务、电子政务中应用最普遍,技术最成熟,可操作性最强的一种电子签名方法。它包括普通数字签名和特殊数字签名,其中普通数字签名用到的算法有 RSA、ElGamal、DES/DSA、椭圆曲线数字签名算法等;特殊数字签名有盲签名、代理签名、群签名、门限签名等。

### 3.5.1 算法和术语

有许多数字签名操作过程中使用的算法都采用公钥算法,用秘密信息对文件签名,用公开信息去验证。这时的签名过程也叫"用私钥加密",验证过程也叫"用公钥解密"。

实际上,并不仅仅只对类似 RSA 这样的公开密钥算法才能完成数字签名过程。也可以使用单向 Hash 算法和时间标记完成签名和验证过程,但是这种情况下对签名和验证过程进行处理要增加额外步骤,而且 Hash 算法可用作数字签名,但不能用作加密,因为它无法解密。

一般地,签名和验证过程通常不包括签名操作所使用的算法细节。其中,签名的生成过程如图 3-5 所示。

相应的验证过程如图 3-6 所示。

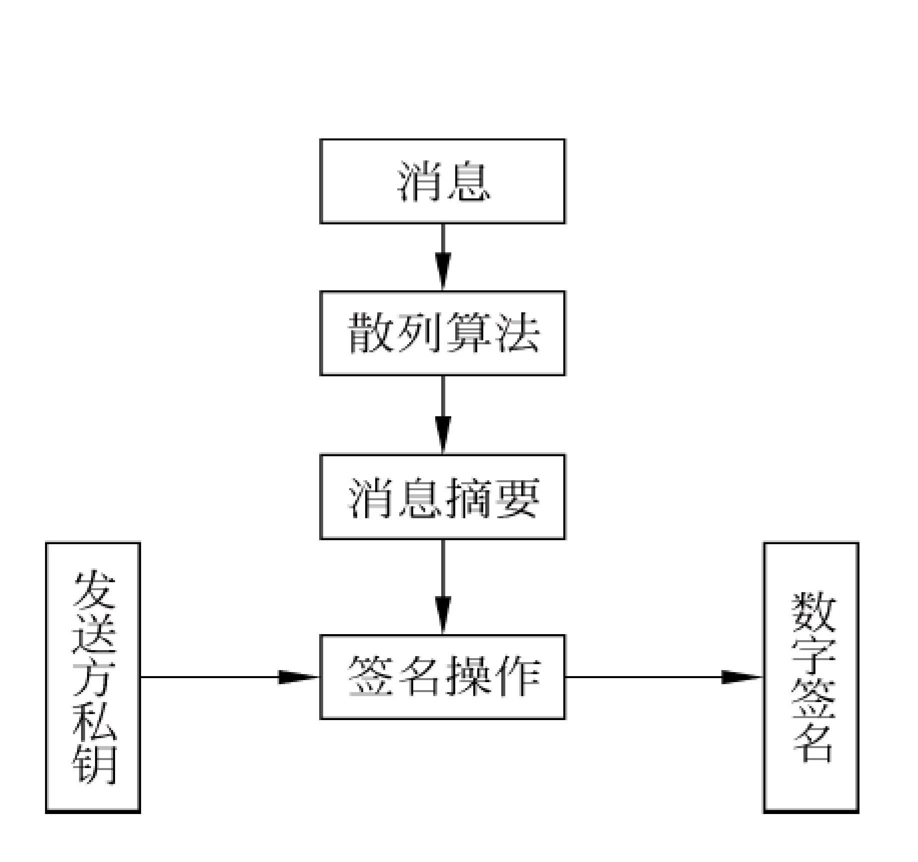


图 3-5 签名的生成过程

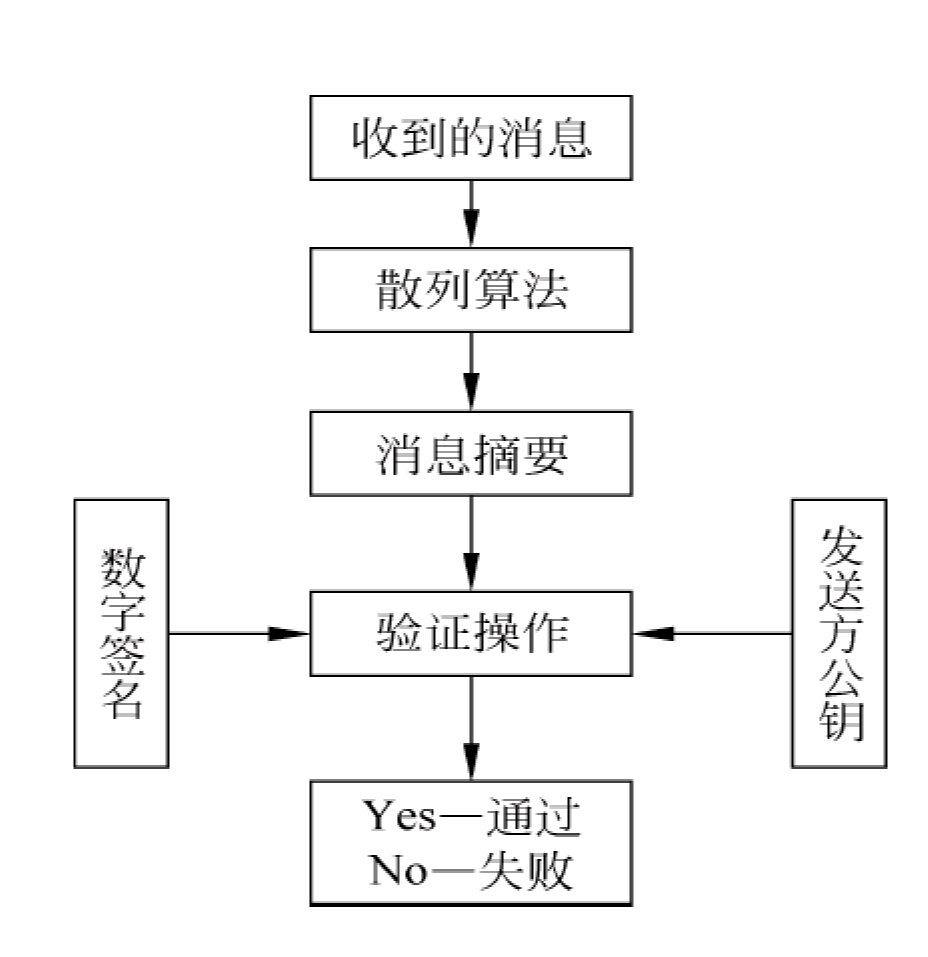


图 3-6 签名的认证过程

更详细地看，数字签名过程原理如图 3-7 和图 3-8 所示。

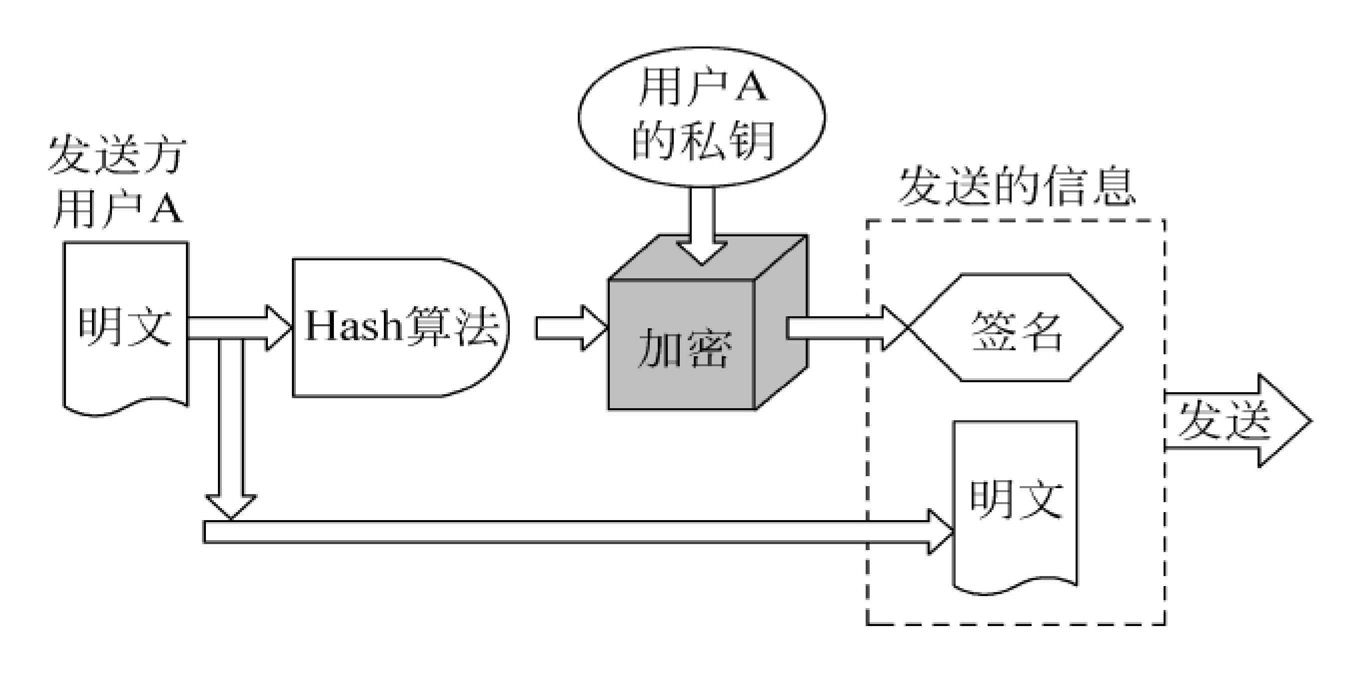



图 3-7　数字签名原理（发送方）

在接收方，为了验证签名的真伪，采用下述方式：用户方接收到的信息中，签名部分采用约定的解密算法，解密出 Hash 值；而明文部分则采用约定的 Hash 算法计算出明文的 Hash 值。将解密所得 Hash 值与计算所得 Hash 进行比较，如果一致，就通过签名验证。

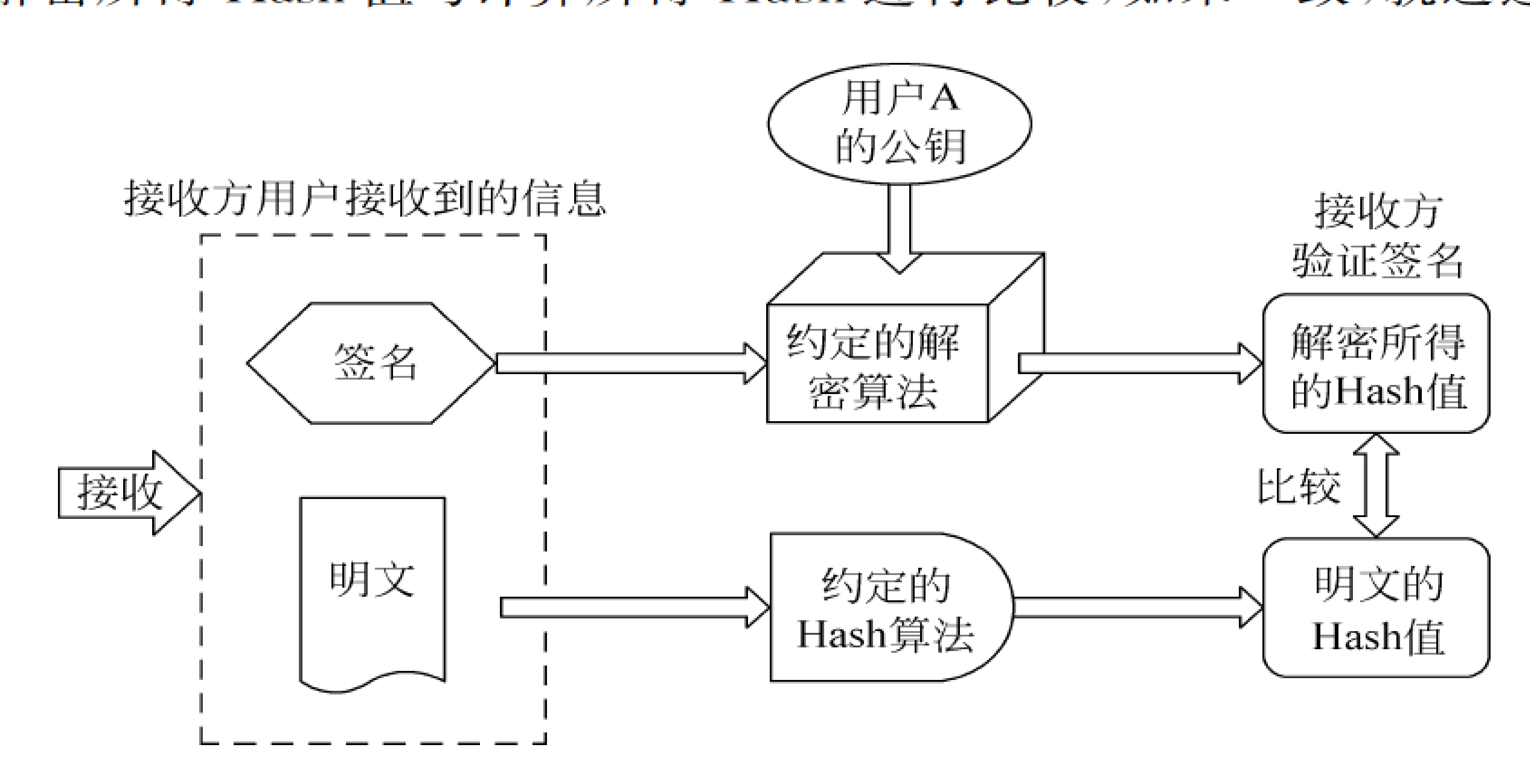



图 3-8　数字签名原理（接收方）

在签名时，附在文件上的比特串叫数字签名（在上面的例子中，用私钥对文件的单向 Hash 值加密）或者叫签名。信息的接收者用以确认发送者的身份和信息的完整性的整个协议叫做认证。关于协议的详细内容将在后面讨论。

## 3.5.2　使用对称密码系统和仲裁者的文件签名

假设 A 想对数字消息签名，送给 B。在 T 和对称密码系统的帮助下，她能做到。

T 是一个有权的、值得依赖的仲裁者。他可以同时与 A 和 B（也可以是其他想对数据文件签名的任何人）通信。他和 A 共享密钥 $K_A$，和 B 共享另一个不同的密钥 $K_B$。并假定这些密钥在协议开始前就已建好，并且为了多次签名可多次重复使用。

签名协议过程如下：

（1）A 用 $K_A$ 加密准备发送给 B 的信息，并把它传送给 T。

（2）T 用 $K_A$ 解密信息。

（3）T 把这个解密信息和他收到 A 信息的声明，一起用 $K_B$ 加密。

(4) T 把加密的信息包传给 B。

(5) B 用 $K_B$ 解密信息包,他就能读 A 所发的信息和 T 的证书,证明信息来自 A。

T 怎么知道信息是从 A 而不是从其他冒名顶替者那里来的呢?答案是,从信息的加密推断出来。由于只有他和 A 共享他们两人的密钥,所以只有 A 能用这个密钥加密信息。

这和文件签名的意义一样吗?让我们看看协议的特点:

(1) 这个签名是可信的,T 是可信的仲裁者,并且知道消息是从 A 那里来的,T 的证书对 B 起着证明的作用。

(2) 这个签名是不可伪造的。只有 A(和 T,因为每个人都相信他)知道 $K_A$,因此只有 A 才能把用 $K_A$ 加密的信息传给 T。如果有人冒充 A,T 在步骤(2)马上就会察觉,并且不会去证明这一步的可靠性。

(3) 这个签名是不能重新使用的。如果 B 想把 T 的证书附到另一个信息上,A 就知道受骗了。仲裁者(可能是 T 或者可存取同一信息的另一名仲裁者)就会要求 B 同时提供信息和 A 加密后的信息,然后仲裁者就用 $K_A$ 加密信息,他马上就会发现它与 B 提供的加密信息不相同。很显然,B 由于不知道 $K_A$,他不可能提供加密信息使它与用 $K_A$ 加密的信息相符。

(4) 签名文件是不能改变的。B 想在接收后改变文件,T 就可用刚才描述的同样办法证明 B 的愚蠢行为。

(5) 签名是不能抵赖的,即使 A 以后声称她没有发信息给 B,T 的证书会说明真相。注意:T 是每个人都信任的,他说的都是正确的。

如果 B 想把 A 签名的文件给 C 阅读,他不能把自己的密钥交给她,他还得通过 T:

(1) B 把信息和 T 关于信息是来自 A 的声明用 $K_B$ 加密,然后送回给 T。

(2) T 用 $K_B$ 解密信息包。

(3) T 检查他的数据库,并确认原始信息是从 A 那里来的。

(4) T 用他和 C 共享的密钥 $K_C$ 重新加密信息包,把它送给 C。

(5) C 用 $K_C$ 解密信息包,她就能阅读信息和 T 证实信息来自 A 的证书。

虽然这些协议是可行的,但对 T 来说是非常耗时的。他不得不整天加密、解密信息,在彼此想发送签名文件的每一对人之间充当中间人。他必须备有数据库信息(虽然可以通过把发送者加密的信息复制发送给接收者来避免)。在任何通信系统中,即使他是毫无思想的软件程序,都会成为通信的“瓶颈”。

更困难的是如何找到像 T 那样的、网络用户都信任的人。T 必须是完美无缺的,即使他在 100 万次签名中只犯了一个错误,也将不会有人再信任他。T 必须是绝对安全的,如果他的密钥数据库泄露了,或有人能修改他的程序代码,所有人的签名可能是完全无用的。一些声称是数年前签名的假文件便可能出现,这将引起混乱,政府可能倒台,混乱状态可能盛行。理论上这种协议或许是可行的,但实际上不能很好运转。

### 3.5.3 数字签名树

Ralph Merkle 提出了一种基于密钥密码的数字签名方案,该方案利用树形结构产生无限多的一次性签名。这个方案的基本思想是在某些公开文档中放入树的根文件,从而

认证它。根节点对一个信息签名,并认证树中的子节点,这些节点的每一个都对信息签名,并对它的子节点认证,一直延续下去。

### 3.5.4　使用公钥密码对文件签名

有几种公钥算法可以用作数字签名。有些算法,例如 RSA,公钥或者私钥都可用作加密。如果用私钥加密文件,就有了安全的数字签名,任何人都可以用对应的公钥来判断这份文件是否为该私钥持有者加的密,也就是所谓的签名。

使用公钥密码签名文件的基本协议是:

(1) A 用自己的私钥对文件加密,从而对文件签名。

(2) A 将签名的文件传给 B。

(3) B 用 A 的公钥解密文件,从而验证签名。

这个协议与以前的算法相比较好,因为不需要 T 去签名和验证,而只需要证明 A 的公钥的确是自己的。甚至连协议的双方都不需要 T 来解决争端。假设 B 不能完成步骤(3),那么他就可以知道签名是无效的。

这个协议也可以满足大家期待的签名特征:

(1) 签名是可信的。当 B 用 A 的公钥验证信息时,就知道是由 A 签名的。

(2) 签名是不可伪造的。只有 A 知道她的私钥。

(3) 签名是不可重用的。签名是文件的函数,并且不可能转换成另外的文件。

(4) 被签名的文件是不可改变的。如果文件有任何改变,文件就不可能用 A 的公钥验证。

(5) 签名是不可抵赖的。B 不用 A 的帮助就能验证 A 的签名。

### 3.5.5　文件签名和时间标记

在上述使用公钥的签名方案中,B 在某些情况下可以欺骗 A。比如,当他把签名和文件一起重用的时候,如果 A 在合同上签名,这种重用不会有什么问题。但如果 A 在一张数字支票上签名,这样做就可能有问题。

设想,假若 A 交给 B 一张 100 美元的签名数字支票,B 把支票拿到银行去验证签名,然后把钱从 A 的账户上转到自己的账上。如果 B 是一个无耻之徒,保存了数字支票的副本。过了一星期,他又把数字支票拿到银行(或者可能是另一个银行),银行验证数字支票并把钱转到他的账上。只要 A 不去对支票本清账,B 就可以一直这样干下去。

因此,与纸质文件上的签名一样,数字签名也经常需要包含时间标记。把对日期和时间的签名附在信息中,并跟信息中的其他部分一起签名。在上面的例子中,银行将时间标记存储在数据库中。这样,当 B 第二次想支取 A 的支票时,银行就要检查时间标记是否和数据库中的一样。由于银行已经从 A 的支票上支付了这一时间标记的支票,就不能通过。

### 3.5.6　用公钥密码和单向 Hash 算法对文件签名

在实际的实现过程中,采用公钥密码算法对长文件签名效率太低。为了节约时间和

成本开销，数字签名协议经常和单向 Hash 算法一起使用。这种情况下，签名者 A 并不是对整个文件签名，而只是对文件的 Hash 值签名。在下面这个协议中，单向 Hash 算法和数字签名算法是事先就协商好的。

(1) A 产生文件的单向 Hash 值。

(2) A 用她的私钥对 Hash 值加密，凭此表示对文件签名。

(3) A 将文件和 Hash 值加密的结果送给 B。

(4) B 用 A 发送的文件产生文件的单向 Hash 值，然后用 A 的公钥对签名的 Hash 值进行解密。如果接收到的签名 Hash 值与自己运算产生的 Hash 值匹配，签名就是有效的。

这样一来计算速度就会大大地提高，并且两个不同的文件有相同的 160 比特 Hash 值的概率为 $1/2^{160}$。因此，使用 Hash 算法的签名和文件签名一样安全。

这个协议还有其他优点。首先，签名和文件可以分开保存。其次，接收者对文件和签名的存储量要求大大降低了。档案系统可用这类协议来验证文件的存在而不需保存它们的内容。中央数据库只存储各个文件的 Hash 值，根本不需要看文件。用户将文件的 Hash 值传给数据库，然后数据库对提交的文件加上时间标记并保存。如果以后有人对某文件的存在发生争执，数据库可通过找到文件的 Hash 值来解决争端。这里可能牵扯到大量的隐秘：A 可能有某文件的版权，但仍保持文件的秘密。只有当她想证明她的版权时，她才不得不把文件公开。

### 3.5.7 多重签名方案

本小节介绍 A 和 B 如何对同一数字文件进行签名。如果不用单向 Hash 算法，有两种选择，如下：

第一种选择是 A 和 B 分别对文件的副本签名，结果签名的信息是原文的两倍。

第二种选择就是 A 首先签名，然后 B 对 A 的签名再进行签名，这是可行的，但是在不验证 B 的签名的情况下就验证 A 的签名是不可能的。

如果采用单向 Hash 算法，则采用下述协议过程进行多重签名：

(1) A 对文件的 Hash 值签名。

(2) B 对文件的 Hash 值签名。

(3) B 将他的签名交给 A。

(4) A 把文件、她的签名和 B 的签名发给 C。

(5) C 验证 A 和 B 的签名。

A 和 B 可以同时或顺序地完成步骤(1)和步骤(2)，到步骤(5)，C 可以只验证其中一人的签名而不用验证另一人的签名。

### 3.5.8 抗抵赖的数字签名

抵赖的情形是指，签名者 A 利用数字签名进行欺骗，她先对文件签名，然后声称并没有签名。

例如，A 先按常规对文件签名，然后却以匿名的形式发布她的私钥，或故意把私钥丢

失在公共场所，或者假装做上面两者中的一种。这样，发现该私钥的任何人都可装成是 A 对文件签名。于是 A 就可以声明她的签名受到侵害，其他人正在假装她签名等。签名者否认对文件的签名和任何其他的用她的私钥签名的文件，这叫做抵赖。

一般采用时间标记可以限制这种欺骗。但实际情况下，故意的抵赖者 A 总可以声称她的密钥在之前就丢失了。如果 A 把事情做得好，她就可以先对文件签名，然后成功地声称并没有对文件签名。这就是为什么要求把私钥隐藏在防拆的模块中，这样一来，A 就不可能接近和乱用私钥了。

虽然没有办法阻止这种可能的乱用，但可以采取措施保证旧的签名不会失效。采用数字签名文件的接收者持有签名的时间标记的方案就能解决这个问题。

一般的协议过程是：

(1) A 对信息签名。

(2) A 产生一个报头，报头中包含有认证信息。A 把报头和签名的信息连接起来，对连接的信息签名，然后把签名的信息发给 T。

(3) T 验证外面的签名，并确认认证信息。T 在 A 签名信息中增加一个时间标记和认证信息。然后对所有的信息签名，并把它发给 B 和 A。

(4) B 验证 T 的签名、认证信息和 A 的签名。

(5) A 验证 T 发给 B 的信息。如果 A 的确没有发起并签名这个信息，她很快就会发现。

另一个方案是在事后有劳 T。B 在接收到签名信息后，可能把副本发给 T 验证，T 能够证实 A 的签名的有效性。

### 3.5.9　数字签名的国际应用

数字签名最早的建议应用之一是用来对禁止核试验条约的验证。美国和前苏联互相允许把地震测试仪放入另一个国家中，以便对核试验进行监控。问题是每个国家需要确信东道国没有篡改从监控国家的地震仪传来的数据。同时，东道主国家需要确信监测器只发送规定的需要监测的信息。

传统的认证技术能解决第一个问题，但只有数字签名能同时解决两个问题。东道国一方只能读，但不能篡改从地震测试仪传来的数据；而监督国则可以确信数据没有被篡改。

## 3.6　本章重点和难点

本章是关于安全协议中需要用到的有关密码学知识，重点在 3.1 节密码学与安全协议的关系、3.3 节利用密码算法建立安全通信信道的原理，以及 3.5 节数字签名原理。

本章的难点是，对数字签名技术的理解。建议对本科生只讲 3.5.1 节、3.5.2 节和 3.5.4 节，研究生视情况扩大到 3.5.5 节和 3.5.6 节。3.5 节中其余部分可用于自学，以扩大阅读范围。

# 习题与思考题

1. 协议的定义是什么？
2. 试述安全协议在信息系统中的作用和意义。
3. 试分析和评价不使用密码算法的安全协议的安全性。
4. 混合密码系统可以解决什么问题？
5. 用形式化语言设计一个使用对称密码技术建立安全通信的协议。
6. 用形式化语言设计一个使用非对称密码技术建立安全通信的协议。
7. 用形式化语言设计一个带重发的数字加密协议。
8. 用形式化语言设计一个重发攻击过程。
9. 试设计采用树形管理方式，使用对称密码系统建立安全通行的协议。
10. 采用对称密码系统，试设计一个可以抗抵赖的数字签名协议。
11. 论述伪随机序列对于安全系统的作用，及其评价准则和意义。

# 第 4 章　基本安全协议

本章介绍基本的安全协议。

本章共有 9 个小节，第 4.1 节介绍安全协议的分类；第 4.2 节介绍密钥交换协议；第 4.3 节介绍认证协议；第 4.4 节介绍认证和密钥交换协议；第 4.5 节介绍多密钥公开密码系统；第 4.6 节介绍秘密分割；第 4.7 节介绍秘密共享；第 4.8 节介绍数据库的密码保护；第 4.9 节是本章重点和难点分析。

在过去二十年中，网络和分布式系统的大量使用给网络的使用者提供了巨大的网络资源，通过网络进行传递和共享的信息越来越多，在网络传输过程中的安全问题也越来越得到更多的关注，尤其是近年来，随着电子商务的发展，对于安全的要求更是迫切，而安全协议在其中起着至关重要的地位。

安全协议的定义：安全协议是建立在密码体制基础上的一种通信协议，计算机网络或分布式系统中的参与者通过安全协议的消息传递，借助于密码算法来达到密钥分配、身份认证、信息保密以及安全地完成电子交易等目的。

安全协议是一种通信协议，它的主要目的是利用密码技术实现网络通信中的密钥分发和身份认证。安全协议是网络通信安全系统的基础，是实现计算机网络安全的关键。然而，大量事实表明，有许多安全协议经过安全专家认真仔细地分析、设计和实现后仍然存在漏洞，有些甚至在使用许多年后才被发现有漏洞。

安全协议的目标分为认证性、非否认性、可追究性、公平性 4 种，其中，认证性应用最为广泛和重要。

## 4.1　安全协议的分类

在计算机网络和分布式系统中，当进行资源访问控制或通信时，一方主体(包括用户、进程、计算机、服务等)需要证实另一方主体的身份，甚至还需要在主体间分配密钥或是其他的秘密信息。安全认证协议就是用来描述主体之间如何证实身份以及分配秘密的，通常由一系列主体之间交换消息组成。认证可能涉及两方或是多方主体(如密钥分配中心等)，可能是单向认证或相互认证，也可能使用对称或是非对称密钥系统。认证协议是网络安全性的基础，即使建立在完善密码系统上的认证协议仍然可能存在各种各样的安全漏洞。

安全协议有多种分类方法，一种方法根据参与者以及密码算法的使用情况进行分类，可以分为 7 类。

(1) 无可信第三方的对称密钥协议。属于这一类的典型协议包括以下 ISO 系列协

议：ISO 对称密钥一遍单边认证协议、ISO 对称密钥二遍单边认证协议、ISO 对称密钥二遍相互认证协议、ISO 对称密钥三遍相互认证协议、Andrew 安全 RPC 协议等。

（2）应用密码校验函数（cryptographic check functions）的认证协议。属于这一类的典型协议包括以下 ISO 系列协议：ISO 应用 CCF 的一遍单边认证协议、ISO 应用 CCF 的二遍单边认证协议、ISO 应用 CCF 的二遍相互认证协议、ISO 应用 CCF 的三遍相互认证协议。

（3）具有可信第三方的对称密钥协议。属于这一类的典型协议包括 Needham-Schroeder（共享密钥型）协议、Otway-Rees 协议、Yahalom 协议、大嘴青蛙协议、Denning-Sacco 协议、Woo-Lam 协议等。

（4）使用对称密钥的签名协议。例如 Needham-Schroeder 签名协议。

（5）使用对称密钥的重复认证协议。属于这一类的典型协议有 Kerberos 协议版本 5、Neuman-Stubble-bine 协议、Kao-Chow 重复认证协议等。

（6）无可信第三方的公钥协议。属于这一类的典型协议包括以下 ISO 系列协议：ISO 公钥一遍单边认证协议、ISO 公钥二遍单边认证协议、ISO 公钥二遍相互认证协议、ISO 公钥三遍相互认证协议、ISO 公钥二遍并行相互认证协议、Diffie-Hellman 协议等。

（7）具有可信第三方的公钥协议。属于这一类的典型协议有 Needham-Schroeder（公开密钥型）协议等。

这种分类方法比较零乱，本书采用下述分类方法，按照协议完成的功能进行划分，可以分成以下 4 类：

（1）密钥交换协议。一般情况下，是在参与协议的两个或者多个实体之间建立共享的密钥，通常用于建立在一次通信中所使用的会话密钥。协议可以采用对称密码体制，也可以采用非对称密码体制，例如 Diffie-Hellman 密钥交换协议。

（2）认证协议。认证协议中包括实体认证（身份认证）协议、消息认证协议、数据源认证协议和数据目的认证协议等，用来防止假冒、篡改、否认等攻击。

（3）认证和密钥交换协议。这类协议将认证和密钥交换协议结合在一起，是网络通信中最普遍应用的安全协议。常见的有 Needham-Schroeder 协议、分布认证安全服务（DASS）协议、ITU-TX. 509 认证协议等。

（4）电子商务协议。与上述协议不同的是，电子商务协议中的参与者往往代表交易的双方，其利益目标是不一致的，甚至根本就是矛盾的。因此，电子商务协议最为关注的就是公平性，即协议应保证交易双方都不能通过损害对方利益而得到它不应得的利益。另外，电子商务协议关心的还有原则性问题。例如 SET 协议等。

## 4.2 密钥交换协议

应用密码算法的前提是通信双方具有相应的加密和解密密钥，对这些密钥进行发布或协商的协议称为密钥交换协议。通常的密码技术是用单独的密钥对每一次单独的会话加密，这个密钥称为会话密钥，因为它只在一次特殊的通信中使用。会话密钥只用于通信期间。这个会话密钥怎么到达会话者的手中是很复杂的事情。

由于存在两种不同的密码体制,对称密码体制和非对称密码体制,不同体制下对密钥的安全性有不同要求,相应的密钥分配协议要求具有不同的安全性质。

### 4.2.1　使用对称密码的密钥交换协议

采用对称密码体制时,通信双方使用同一个钥码进行信息的加密和解密。密钥交换协议的目的是使通信双方共同拥有这个密钥。因而,对密钥交换协议的安全要求包括以下内容。

① 秘密性:保证非法主体不能获知该密钥。

② 完整性:保证双方拥有的是同一个密钥。

③ 可用性:保证密钥交换协议最终可以执行结束,并且在协议执行结束后双方可以建立起密钥。

这类协议假设网络上的用户 A 和 B 每人与密钥分配中心(KDC)共享一个密钥,下面的协议中 KDC 用 T 表示。

在协议开始执行前,这些密钥必须在适当的位置。在这里,协议忽略了怎么使密钥处在适当的位置,即怎么分配这些密钥这个非常实际的问题,只是假设它们已经在适当的位置,并且 M 不知道。

(1) A 呼叫 T,并请求一个与 B 通信的会话密钥。

(2) T 产生一随机会话密钥,并对它的两个副本加密:一个用 A 的密钥,另一个用 B 的密钥加密,T 发送这两个副本给 A。

(3) A 对她的会话密钥的副本解密。

(4) A 将 B 的会话密钥副本送给 B。

(5) B 对他的会话密钥的副本解密。

(6) A 和 B 用这个会话密钥安全地通信。

这个协议依赖于 T 的绝对安全性。T 可能是可信的计算机程序,而不是可信的个人。

如果 M 破坏了 T,整个网络都会遭受损害:他有 T 与每个用户共享的所有密钥;他可以读所有过去和将来的通信业务。他所做的事情就是对通信线路进行搭线窃听,并监视加密的报文业务。

这个系统的另一个问题是 T 可能会成为“瓶颈”。他必须参与每一次密钥交换,如果 T 失败了,这个系统就会被破坏。

### 4.2.2　使用公开密钥密码的密钥交换协议

采用非对称密码体制时,通信双方各拥有一对密钥,称为公开密钥和私有密钥。公开密钥可以向外界公布,由其他协议参与者用来对消息进行加密、解密或签名验证;私有密钥不对外公开,由密钥所属者用来相应地对消息进行解密、加密或签名。此时所谓的密钥分配是指对公开密钥进行发布。密钥分配协议的目的是使通信双方彼此知道对方的公开密钥,因而,对密钥分配协议的安全要求如下。

① 完整性:保证接收方收到的密钥确实是发送方的公开密钥。

② 可用性：保证密钥分配协议最终可以执行结束，并且通信双方最终可以得到对方的公开密钥。

在这类协议中，A 和 B 使用公开密钥密码协商会话密钥，并用协商的会话密钥加密数据。在一些实际应用中，A 和 B 签了名的公开密钥可在数据库中获得。这使得密钥交换协议更容易，即使 B 从来没有听说过 A，A 也能够把信息安全地发送给 B。

(1) A 从 KDC 得到 B 的公开密钥。

(2) A 产生随机会话密钥，用 B 的公开密钥加密它，然后传给 B。

(3) B 用自己的私钥解密 A 的信息。

(4) A、B 两人用同一会话密钥对他们的通信进行加密。

## 4.3 认证协议

当 A 登录计算机(或自动柜员机、电话银行系统，或其他的终端类型)时，计算机怎么知道并确认她是谁呢？这需要用认证协议来解决。

传统的办法是用通行字来解决这个问题。A 先输入她的通行字，然后计算机确认这个通行字是正确的，从而推断出登录的人是 A。A 和计算机两者都知道这个秘密通行字，而且 A 每次登录时，计算机都要求 A 输入通行字。

### 4.3.1 利用单向函数的认证

实际上，在使用的时候，没有必要让计算机存储通行字。计算机只要有能力区别有效通行字和无效通行字就行。

这种思路可以用单向函数来实现。计算机存储通行字的单向函数而不是存储通行字。这样，认证过程如下：

(1) A 将她的通行字传送给计算机。

(2) 计算机完成通行字的单向函数计算。

(3) 计算机把单向函数的运算结果和它以前存储的值进行比较。

这样，由于计算机不再存储每人的有效通行字表，所以某些人侵入计算机，并偷取通行字的威胁就减少了。由通行字的单向函数生成通行字表是没用的，因为单向函数不可能逆向恢复出通行字。

但是，用单向函数加密的通行字文件还是脆弱的。例如，M 可以在业余时间编制 100 万个最常用的通行字表，用单向函数对所有 100 万个通行字进行运算，并将结果存储起来。如果每个通行字大约是 8 个字节，运算结果的文件不会超过 8MB。

现在 M 偷出加密的通行字文件。把加密的通行字和已加密的可能通行字文件进行比较，再观察哪个能匹配。这就是字典式攻击，它的成功率非常高。Salt 是使这种攻击更困难的一种方法。

Salt 是一种随机字符串，与通行字连接在一起，再用单向函数对其运算。然后将 Salt 值和单向函数运算的结果存入主机数据库中。

如果 Salt 值可能的数目足够大，实际上就消除了对常用通行字采用的字典式攻击。

因此 M 不得不产生每个可能的 Salt 值的单向 Hash 值。这是初始化矢量的简单尝试。

这里的关键是当 M 试图破译其他人的通行字时，要迫使他不得不每次试验字典里的每个通行字的加密，而不是只对可能的通行字进行预先计算。

使用较大的 Salt 是必需的，大多数 Unix 系统仅使用 12 比特的 Salt。但是，DanielKlein 开发了一个猜测通行字的程序，在大约一星期的时间里，经常能破译出一个给定的系统中的 40%的通行字。DavidFeld-meier 和 PhilipKarn 编辑了大约 732 000 个常用的通行字表，表中的通行字都和 4096 个可能的 Salt 值中的每个值有联系。采用这张表，他们估计在一给定系统中，大约能够破译出 30%的通行字。

Salt 不是万灵药，增加 Salt 的比特数不能解决所有问题。Salt 只能防止对通行字文件采用的一般的字典式攻击，不能防止对通行字的一致攻击。通常情况下，有不少人习惯在多个机器上有相同的通行字，这样做比选用拙劣的通行字更为危险。

### 4.3.2 SKEY 认证

SKEY 是一种认证程序，又称为一次性通行字系统（one-time password system），依赖于单向函数的安全性，采用 MD4 或 MD5 算法。

在认证之前，需要先配置系统，配制方法是：A 输入随机数 R，以及使用次数 n，计算机计算通行字 f(R)、f(f(R))、f(f(f(R)))等，假如计算次数 $n=100$ 次。分别用 $x_1, x_2, x_3, \cdots, x_{100}$ 来表示。其中，$x_{100}=f(R)$，$x_{99}=f(x_{100})$，$x_{98}=f(x_{99})$等。

计算机打印出这些数的列表，A 把它们妥善保管。计算机在登录数据库中 A 的名字后面存储 100 个计算出来的值。

当 A 第一次登录时，输入她的名字和 $x_{100}$，计算机计算 $f(x_{100})$，并把它和 $x_{99}$ 比较，如果它们匹配，那么可以证明 A 的身份是真的。然后，计算机更新数据库，消去 $x_{100}$。同时 A 也从她的列表中消去 $x_{100}$。

这样，A 在第 i 次登录时，输入她的列表中未取消的最后的数 $x_{100-i+1}$，而计算机则计算 $f(x_{100-i+1})$，并和存储在它的数据库中的 $x_{100-i}$ 比较。如果匹配，则完成认证，计算机和 A 都从各自的数据中消去 $x_{100-i+1}$；如果不匹配，则认证失败。

因为每个数 $x_i$ 只被用一次，并且采用的函数是单向的，所以 E 不可能得到任何有用的信息。同样，数据库对攻击者也毫无用处。

当然，当 A 用完了列表上面的数据后，必须重新初始化系统。

### 4.3.3 采用公开密钥密码的认证

严格地说，采用通行字方案登录计算机进行认证的协议有严重的安全问题。当 A 将她的通行字发给主机时，能够进入数据通道的任何人都可读取她的通行字。如此一来，就可以访问她的主机，在信道的任何一点 E 都有可能窃听 A 的登录序列。如果 E 可以存取主机的处理机存储器，那么在主机对通行字进行 Hash 运算前，E 都能够看到通行字。

利用公开密钥密码可以解决这个问题。主机保存每个用户的公开密钥文件，所有用户保存自己的私钥。这里给出一个协议。当登录时，协议按下面进行：

（1）主机发送一个随机字符串给 A。

(2) A 用她的私钥对此随机字符串加密,并将此字符串和她的名字一起传送回主机。

(3) 主机在它的数据库中查找 A 的公开密钥,并用公开密钥解密。

(4) 如果解密后的字符串与主机在步骤(1)中发送给 A 的字符串匹配,则允许 A 访问系统。

这样就没有其他人能得到 A 的密钥,因此也就不可能有任何人冒充 A。更重要的是,A 绝不会在传输线路上将她的私钥发送给主机。窃听这个交互过程中的 E,也就不可能得到任何信息来推导出 A 的私钥,从而冒充 A。

因为私钥既长又难记,需要由用户的硬件或通信软件自动处理。这就需要一种 A 信任的智能终端。但主机和通信线路都不必是安全的。

安全的身份证明协议采用下面更复杂的形式:

(1) A 根据一些随机数和她的私钥进行计算,并将结果传送给主机。

(2) 主机将一不同的随机数传送给 A。

(3) A 根据这些随机数(A 产生的和从主机接收的)和她的私钥进行一些计算,并将结果传送给主机。

(4) 主机用从 A 那里接收来的各种数据和 A 的公开密钥进行计算,以此来验证 A 是否知道自己的私钥。

(5) 如果 A 知道,则她的身份就被证实了。

如果 A 不相信主机,就像主机不相信她一样,那么 A 将要求主机用同样方式证实其身份。

步骤(1)似乎是不必要的,但它可以用来阻止对协议的攻击。

### 4.3.4 用连锁协议互相认证

假设 A 和 B 是想要互相认证的两个用户。他们每人有一个另一人知道的通行字:A 的通行字是 $P_A$,B 的通行字是 $P_B$。下面的协议是行不通的:

(1) A 和 B 交换公开密钥。

(2) A 用 B 的公开密钥加密 $P_A$,并将它传送给 B。

(3) B 用 A 的公开密钥加密 $P_B$,并发送给 A。

(4) B 解密他在步骤(2)中接收的信息,并验证它是否正确。

(5) A 解密她在步骤(3)中接收的信息,并验证它是否正确。

因此 M 可以成功发起“中间人攻击”,这是一种在通信双方之间进行恶意的主动攻击以此截取和破坏正常的信息交互的方式。下面是攻击过程:

(1) A 和 B 交换公开密钥。M 截取这两个报文,他用自己的公开密钥代替 B 的,并将它发送给 A。然后,他又用自己的公开密钥代替 A 的,并将它发送给 B。

(2) A 用 B 的公开密钥对 $P_A$ 加密,并发送给 B。M 截取这个报文,用自己的私钥对 $P_A$ 解密,再用 B 的公开密钥加密,并将它发送给 B。

(3) B 用 A 的公开密钥对 $P_B$ 加密,并发送给 A。M 截取它,用自己的私钥对 $P_B$ 解密。再用 A 的公开密钥对它加密,并发送给 A。

(4) A 对 $P_B$ 解密,并验证它是正确的。

(5) B 对 $P_A$ 解密,并验证它是正确的。

从 A 和 B 处看并没有什么不同,然而 M 知道 $P_A$ 和 $P_B$。如果 A 是用户,B 是主机,M 可以假装是 B 和 A 一起完成协议的开始几步,然后终止连接。

实施过程要求 M 通过模拟线路噪声或网络失败来终止连接,最终结果是 M 得到了 A 的通行字。然后,M 再和 B 连接,完成协议。这样,M 也就有 B 的通行字了。

假如用户的通行字比主机的通行字更敏感,就可以修改这个协议,使 A 给出自己的通行字之前,让 B 先给出自己的通行字。这样修改后会导致一个更加复杂的攻击过程,不再赘述。

### 4.3.5 SKID 协议

SKID2 和 SKID3 是为 RACE(欧洲高级通信技术与开发)的 RIPE(完整性基本评估)项目开发的对称密码识别协议。它们都采用 MAC 来提供安全性,并且 SKID2 和 SKID3 两个协议都假设 A 和 B 共享同一密钥 K。

在 SKID2 中,允许 B 向 A 证明他的身份。下面是这个协议的过程:

(1) A 选用随机数 $R_A$(RIPE 文件规定 64 比特的数),并将它发送给 B。

(2) B 选用随机数 $R_B$(RIPE 文件规定 64 比特的数),将下面的数发送给 A。

$$R_B, H_K(R_A, R_B, B) \rightarrow A$$

(3) $H_K$ 是 MAC(RIPE 文件建议的 RIPE-MAC 函数),B 是 B 的名字。

(4) A 计算 $H_K(R_A, R_B, B)$,并和她从 B 那里接收到的信息比较,如果结果一致,那么 A 知道她正与 B 通信。

SKID3 提供了 A 和 B 之间的相互认证。步骤(1)~步骤(3)与 SKID2 是一样的,以后的协议按下面进行:

(1) A 向 B 发送。

(2) $H_K(R_B, A) \rightarrow B$。

(3) B 计算 $H_K(R_B, A)$,并将它与从 A 那里收到的信息比较,如果相同,那么 B 相信他正与 A 通信。

这个协议对中间人攻击来说是不安全的。一般地,中间人攻击能够击败任何不包括某些秘密的协议。

### 4.3.6 信息认证

采用信息认证时,需要解决的问题是,当 B 从 A 那里接收信息,他怎么判别接收到的信息是可信的呢?解决这个问题的一个办法是使用签名技术,如果 A 对自己的信息进行签名,就能够使 B 相信信息的来源。

采用对称密码算法可以提供一种认证。当 B 从 A 那里接收到用他们的共享密钥加密的信息时,B 就知道信息是从 A 那里来的,因为没有其他人知道他们的密钥。

然而,要使第三者 T 相信这个事实,这样做还不够;因为 B 不能(或者不愿)把信息给 T 看,并使他相信这个信息是从 A 那里来的。T 能够确认的是,信息的确是从 A 或 B 那里来的,因为没有其他人共享他们的密钥,但是 T 却没有办法判断信息到底是从谁那

里来的。

如果信息没有被加密,A 也可以使用 MAC。这也可以使 B 相信信息是可信的,但存在与上述对称密码的解决方法相同的信息来源确认问题。

# 4.4 认证和密钥交换协议

本节介绍几种认证和密钥交换协议,这些协议综合利用密钥交换和认证,要解决一般的计算机问题:A 和 B 分别坐在网络的两端,他们想安全地交谈。A 和 B 怎么交换密钥呢?他们中的每个人怎么确信他们当时正在同对方交谈而不是同 M 谈话呢?

本节要介绍的大多数协议假设 T 与参与者双方各共享一个不同的密钥,并且所有这些密钥在协议开始前都在适当的位置。在这些协议中使用的符号如表 4-1 所示。

**表 4-1 在认证和密钥交换协议中使用的符号**

| 符　号 | 意　义 |
|---|---|
| $E_A$ | 用 T 和 A 共享的密钥加密 |
| $E_B$ | 用 T 和 B 共享的密钥加密 |
| I | 索引号 |
| K | 随机会话密钥 |
| L | 生存期 |
| $T_A$,$T_B$ | 时间标记 |
| $R_A$,$R_B$ | 随机数,分别由 A 和 B 选择 |

## 4.4.1 简单对称密钥管理协议

这一协议也称为大嘴青蛙(wide-mouth frog)协议,可能是最简单的对称密钥管理协议,该协议使用一个可信的服务器。

假设 A 和 B 两人各与 T 共享一个密钥。这些密钥只作密钥分配用,而不是用作加密用户之间的实际报文。会话密钥只通过两个报文就从 A 传送给 B,传送过程如下:

(1) A 将时间标记 $T_A$ 连同 B 的名字和随机会话密钥 K 一起,用她和 T 共享的密钥对整个报文加密。她将加了密的报文和她的身份 A 一起发送给 T。

$$A, E_A(T_A, B, K) \rightarrow T$$

(2) T 解密从 A 来的报文。然后将一个新的时间标记 $T_B$ 连同 A 的名字和随机会话密钥一起,用他与 B 共享的密钥对整个报文加密,并将它发送给 B。

$$E_B(T_B, A, K) \rightarrow B$$

该协议过程如图 4-1 所示。

在这个协议中,最重要的假设是 A 能够产生好的会话密钥。但是"好的"密钥,即随机的密钥数是不容易产生的。所以,实际上无法相信 A 能够做好这件事。下面这个协议由 T 来产生密钥。

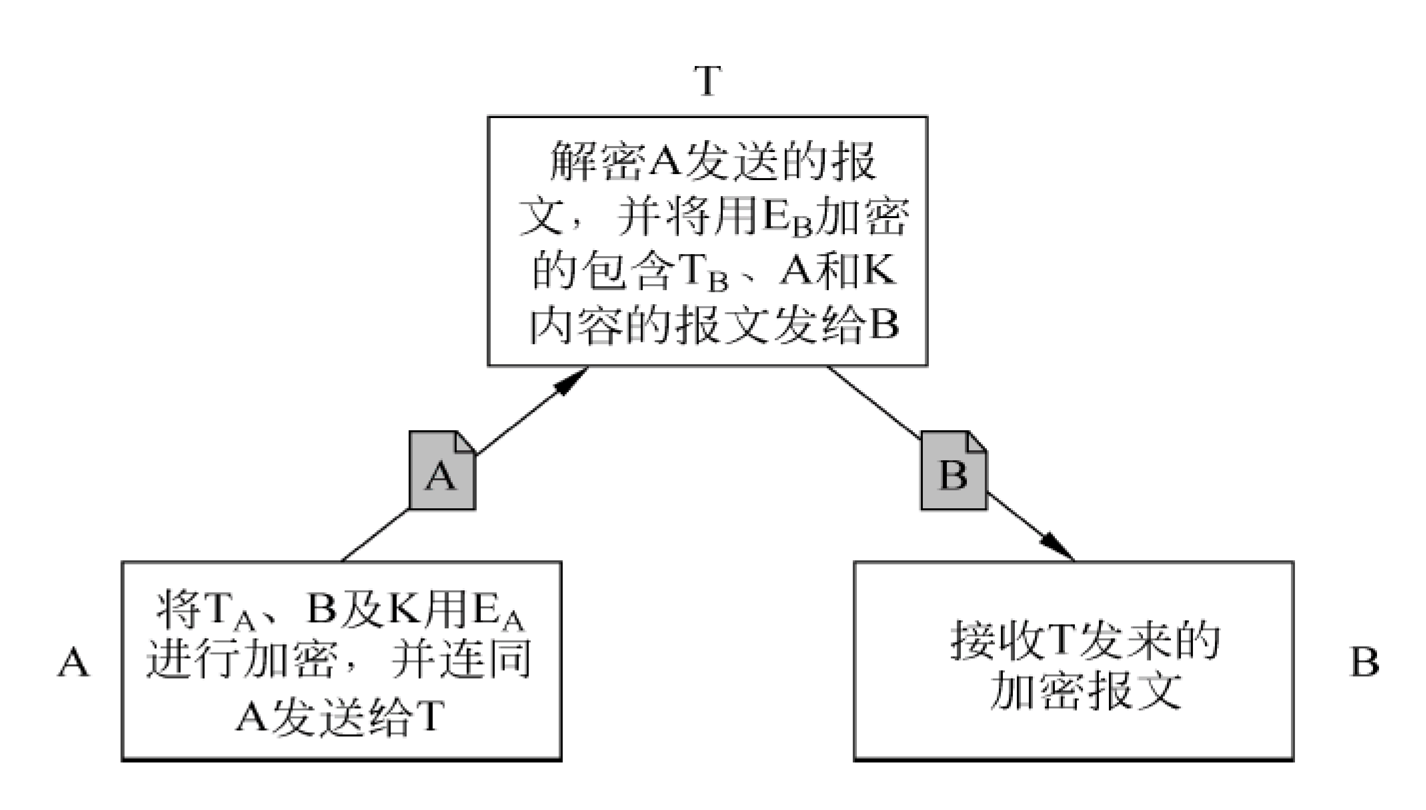



图 4-1　简单对称密钥管理协议

## 4.4.2　带随机数的对称密钥管理协议

这个协议也称为 Yahalom 协议，在这个协议中，假定 A 和 B 两人各与 T 共享一个密钥。

(1) A 将她的名字连同随机数 $R_A$ 一起发送给 B。

$$A, R_A \rightarrow B$$

(2) B 将 A 的名字、A 的随机数、他自己的随机数 $R_B$ 一起用他和 T 共享的密钥加密。再将加密的结果和 B 的名字一起发送给 T。

$$B, E_B(A, R_A, R_B) \rightarrow T$$

(3) T 产生两个报文，第一个报文由 B 的名字、随机会话密钥 K、A 的随机数和 B 的随机数组成。用他和 A 共享的密钥对所有第一个报文加密；第二个报文由 A 的名字和随机会话密钥组成，用他和 B 共享的密钥加密，然后将这两个报文发送给 A。

$$E_A(B, K, R_A, R_B), E_B(A, K) \rightarrow A$$

(4) A 解密第一个报文，提出 K，并确认 $R_A$ 的值与她在步骤(1)中的值一样。A 发送两个报文给 B。第一个报文是从 T 那里接收到的用 B 的密钥加密的报文，第二个报文是用会话密钥加密的 $R_B$。

$$E_B(A, K), E_K(R_B) \rightarrow B$$

(5) B 用他的密钥解密报文，提取 K，并确认 $R_B$ 与他在步骤(2)中的值一样。

该协议过程如图 4-2 所示。

结果是 A 和 B 互相确认正在同对方谈话，而不是跟第三者。这里的新内容是：B 是同 T 接触的第一人，而 T 仅发送给 A 一个报文。

## 4.4.3　带随机数的对称密钥协议的改进

这个协议由 Roger Needham 和 Michael Schroeder 发明，也采用对称密码和 T。协议过程如下：

(1) A 将由她的名字 A，B 的名字 B 和随机数 $R_A$ 组成的报文传给 T。

$$(A, B, R_A) \rightarrow T$$

(2) T 产生一随机会话密钥 K。他用与 B 共享的密钥对随机会话密钥 K 和 A 名字

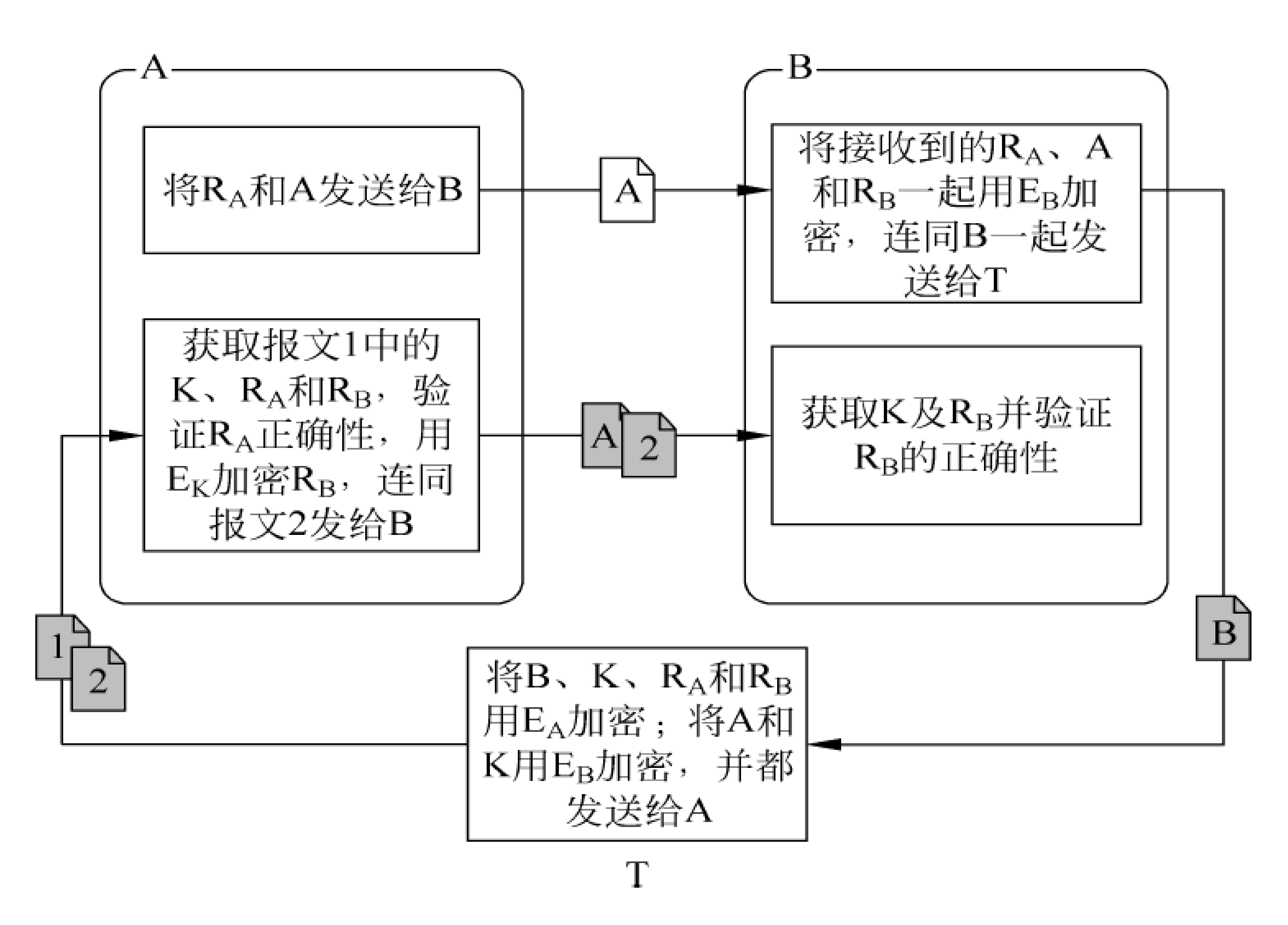


图 4-2 带随机数的对称密钥管理协议

组成的报文加密。然后用他和 A 共享的密钥对 A 的随机数、B 的名字、会话密钥 K 和已加密的报文进行加密，最后，将加密的报文传送给 A。

$$E_A(R_A, B, K, E_B(K, A)) \rightarrow A$$

(3) A 将报文解密并提取 K。她确认 $R_A$ 与她在步骤(1)中发送给 T 的一样。然后她将 T 用 B 的密钥加密的报文发送给 B。

$$E_B(K, A) \rightarrow B$$

(4) B 对报文解密并提取 K，然后产生另一随机数 $R_B$。他用 K 加密它并将它发送给 A。

$$E_K(R_B) \rightarrow A$$

(5) A 用 K 将报文解密，产生 $R_B-1$ 并用 K 对它加密，然后将报文发回给 B。

$$E_K(R_B-1) \rightarrow B$$

(6) B 用 K 对信息解密，并验证它是 $R_B-1$。

该协议过程如图 4-3 所示。

所有这些围绕 $R_A$、$R_B$、$R_B-1$ 的麻烦用来防止重放攻击。在重放攻击中，M 可以记录旧的报文，在以后再使用它们以达到破坏协议的目的。在步骤(2)中 $R_A$ 的出现使 A 确认 T 的报文是合法的，并且不是以前协议的重放。在步骤(5)中，当 A 成功地解密 $R_B$，并将 $R_B-1$ 送回给 B 之后，B 确认 A 的报文不是早期协议执行的重放。

这个协议的主要安全漏洞是旧的会话密钥仍有价值。如果 M 可以存取旧的密钥 K，他可以发起一次成功的攻击。他所做的全部工作是记录 A 在步骤(3)发送给 B 的报文。然后，一旦他有 K，他可以假装是 A：

(1) M 发送给 B 下面的信息。

$$E_B(K, A) \rightarrow B$$

(2) B 提取 K，产生 $R_B$，并发送给“A”。

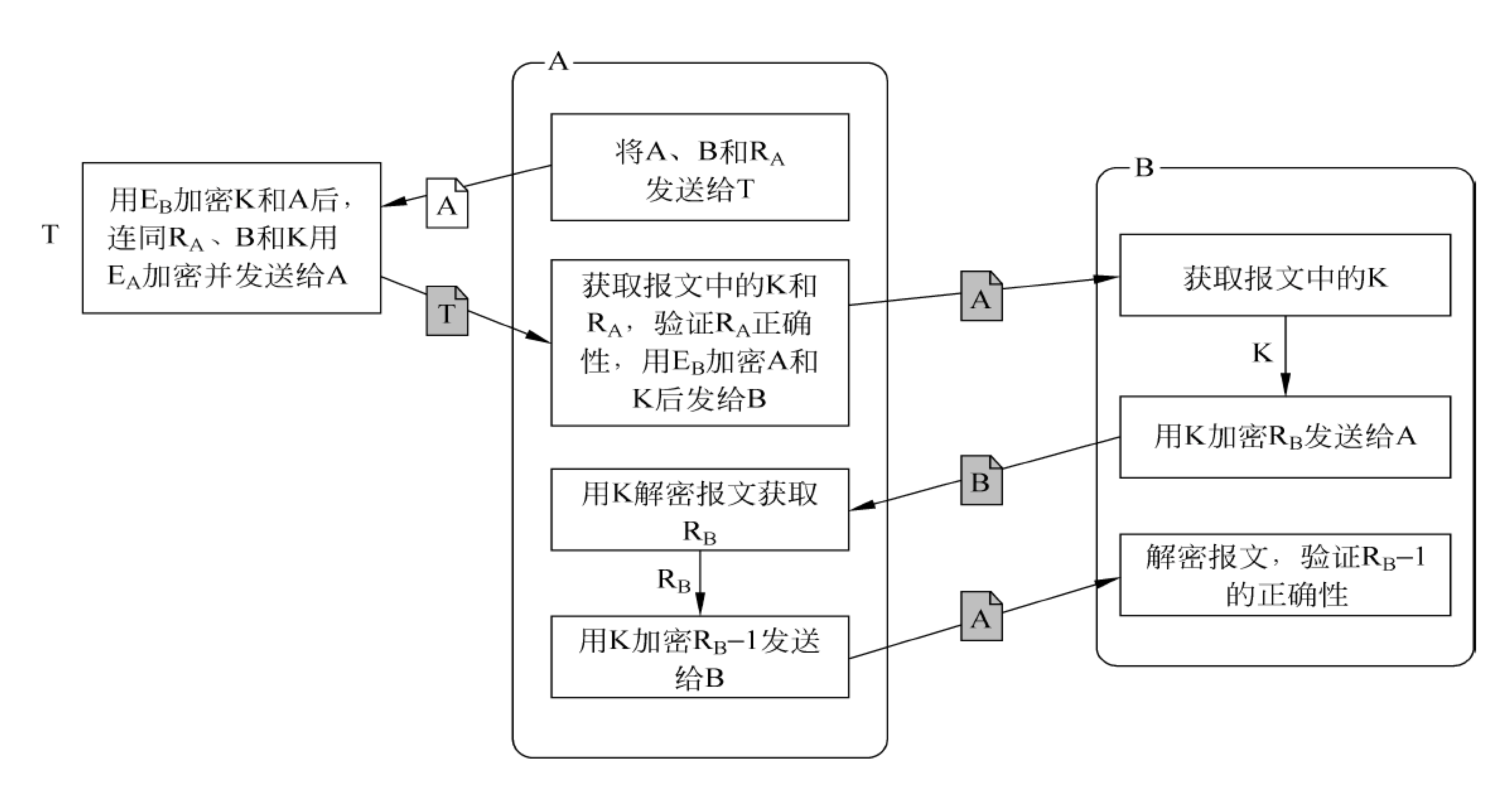



图 4-3　改进的带随机数的对称密钥协议

$$E_K(R_B)\rightarrow A$$

（3）M 截取此报文，用 K 对它解密，并发送给 B。

$$E_K(R_B-1)\rightarrow B$$

（4）B 验证“A”的报文是 $R_B-1$。

该协议过程如图 4-4 所示。

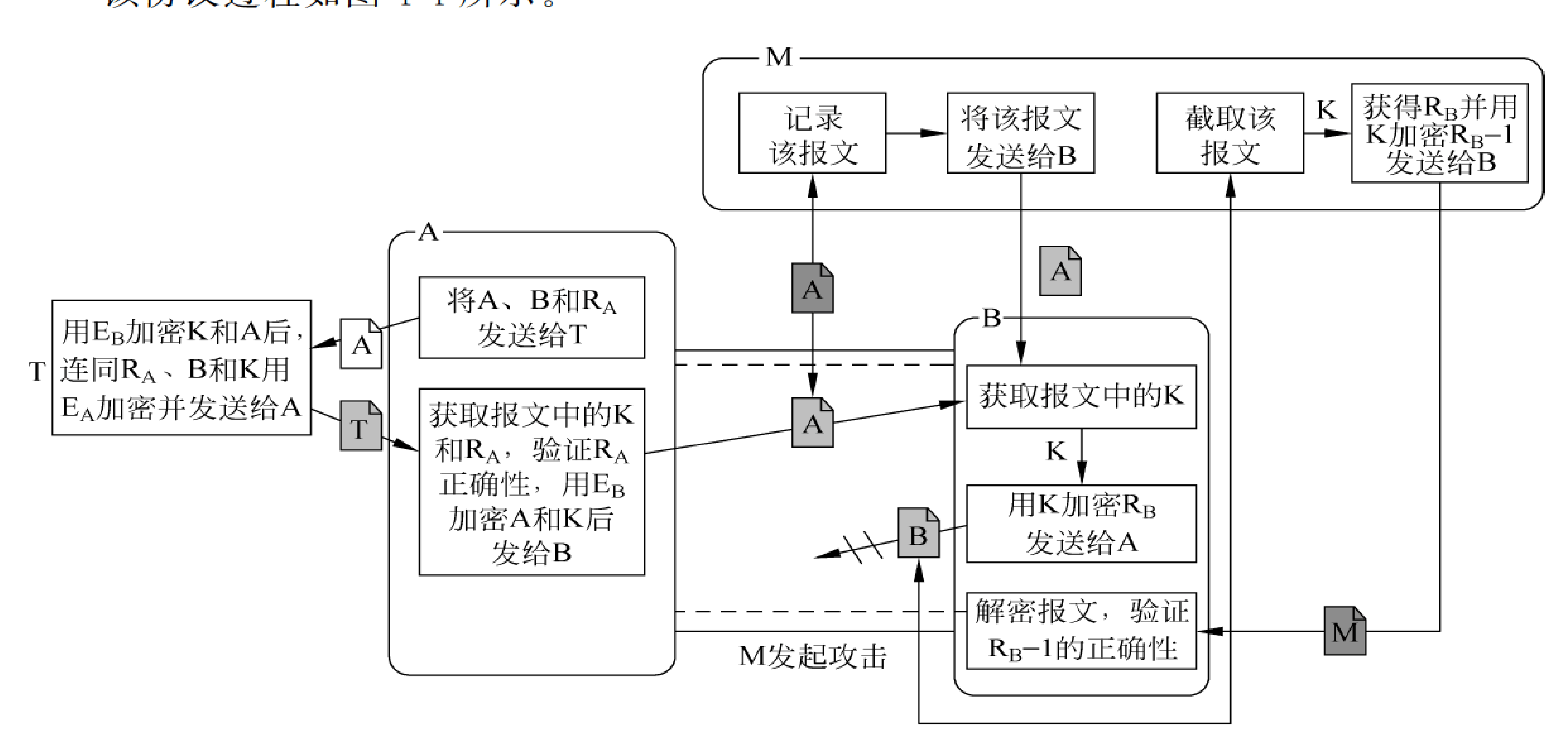



图 4-4　对改进的带随机数的对称密钥协议的攻击

到此为止，M 成功地使 B 确认他就是 A 了。

采用一种更强的使用时间标记的协议能够击败这种攻击。在步骤(2)中，一个时间标

记被附到用 B 的密钥加密的 T 的信息中：$E_B(K,A,T)$。时间标记需要一个安全的和精确的系统时钟，这对系统本身来说不是一个普通问题。

如果 T 与 A 共享的密钥 $K_A$ 泄露了，后果是非常严重的。M 能够用它获得同 B 交谈的会话密钥，或他想要交谈的其他任何人的会话密钥。甚至在 A 更换她的密钥后 M 还能够继续做这种事情。

## 4.4.4 带索引的对称密钥协议

这个协议也称为 Otway-Rees 协议，它也使用对称密码。主要过程如下：

(1) A 产生一个报文，此报文包括一个索引号 I、她的名字 A、B 的名字和一随机数 $R_A$，用 A 和 T 共享的密钥 $E_A$ 对此报文加密，A 将索引号、A 的名字和 B 的名字与加密的报文一起发送给 B。

$$I,A,B,E_A(R_A,I,A,B)\rightarrow B$$

(2) B 产生一个报文，此报文包括一个新的随机数 $R_B$、索引号 I、A 的名字和 B 的名字。用他与 T 共享的密钥 $E_B$ 对此报文加密。他将 A 的加密报文、索引号、A 的名字、B 的名字与他加了密的报文一起发送给 T。

$$I,A,B,E_A(R_A,I,A,B),E_B(R_B,I,A,B)\rightarrow T$$

(3) T 产生一个随机会话密钥 K，然后，产生两个报文。一个是用他与 A 共享的密钥对 A 的随机数和会话密钥加密，另一个是用与 B 共享的密钥对 B 的随机数和会话密钥加密。他将这两个报文与索引号一起发送给 B。

$$I,E_A(R_A,K),E_B(R_B,K)\rightarrow B$$

(4) B 用 A 的密钥加密的报文连同索引号一起发送给 A。

$$I,E_A(R_A,K)\rightarrow A$$

(5) A 解密报文，恢复出她的密钥和随机数，然后她确认协议中的索引号和随机数都没有改变。

该协议过程如图 4-5 所示。

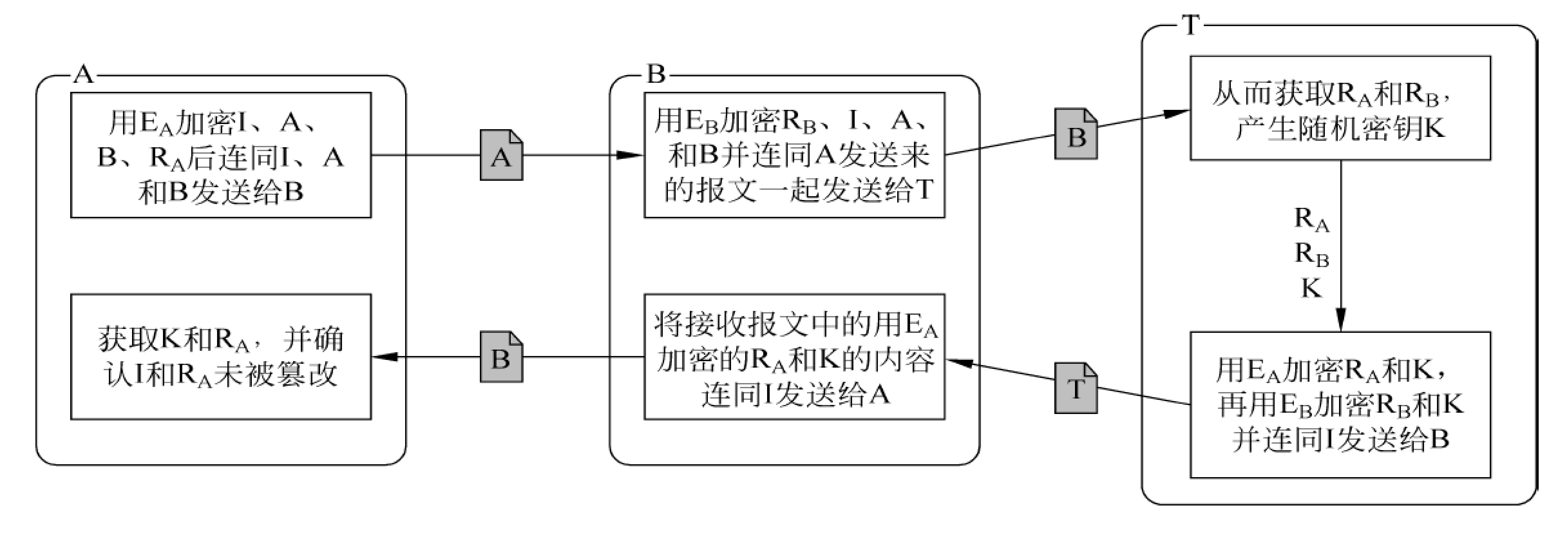



图 4-5 带索引的对称密钥协议

假设所有随机数都匹配，并且按照这种方法索引号没有改变，A 和 B 现在相互确认对方的身份，他们就有一个密钥用于通信了。

## 4.4.5　带时间标记的对称密钥协议

这一协议也称为 Kerberos 协议，是一种 Needham-Schroeder 协议，面向开放系统的，可以为网络通信提供可信第三方服务的认证机制。

在基本的 Kerberos 第 5 版本的协议中，T 和 A、B 中的每人各共享一个密钥。A 想产生一个会话密钥用于与 B 通信。下面是协议内容：

（1）A 将她的身份和 B 的身份发送给 T。

$$A,B\rightarrow T$$

（2）T 产生一个报文，该报文由时间标记 T、使用寿命 L、随机会话密钥 K 和 A 的身份构成。T 用与 B 共享的密钥加密这个报文。然后，他再取时间标记、使用寿命、会话密钥和 B 的身份，并且用他与 A 共享的密钥加密，并把这两个加密报文发给 A。

$$E_A(T,L,K,B),E_B(T,L,K,A)\rightarrow A$$

（3）A 用她的身份和时间标记产生报文，并用 K 对它进行加密，将它发送给 B。A 也将从 T 那里来的用 B 的密钥加密的报文发送给 B。

$$E_K(A,T),E_B(T,L,K,A)\rightarrow B$$

（4）B 用 K 对时间标记加 1 的报文进行加密，并将它发送给 A。

$$E_K(T+1)\rightarrow A$$

该协议过程如图 4-6 所示。

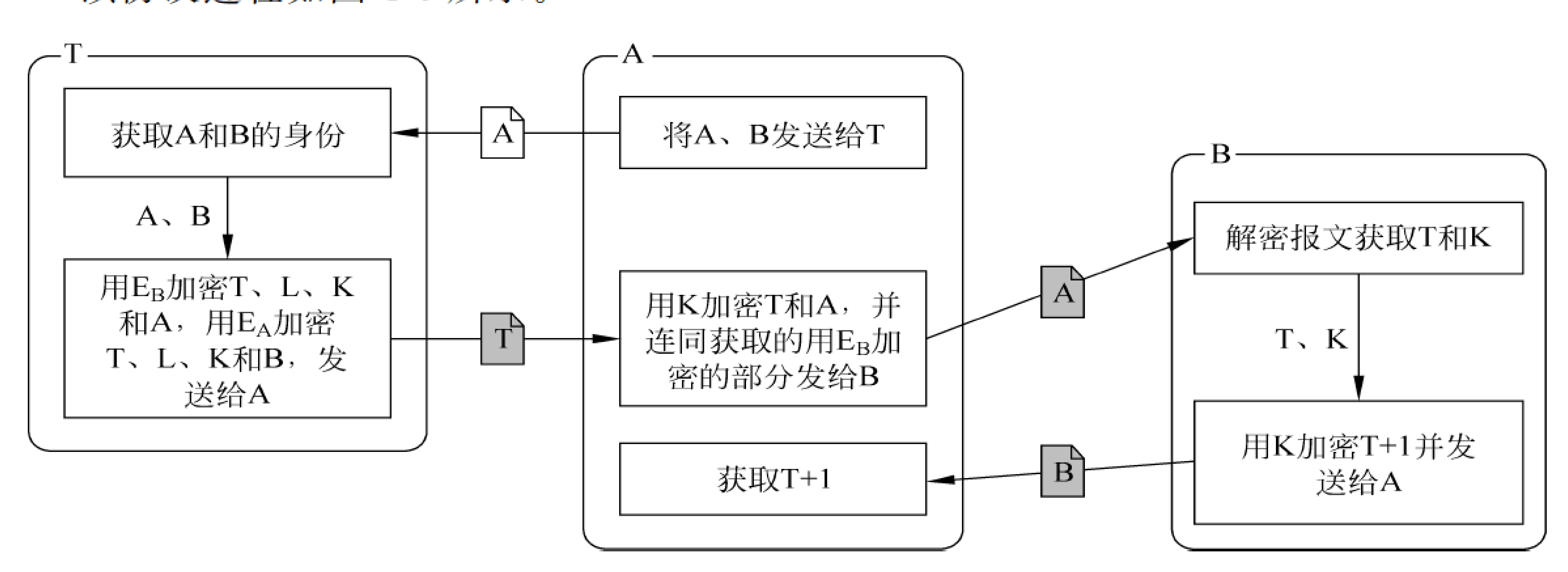



图 4-6　带时间标记的对称密钥协议

这个协议是可行的。但它假设了每个人的时钟都与 T 的时钟同步。实际上，这个结果是通过把时钟同步到一个安全的定时服务器的几分钟之内，并在这个时间间隔内检测重放而获得的。

## 4.4.6　带时间标记和同步的协议

这个协议也称为 Neuman-Stubblebine 协议。不管是由于系统缺陷还是由于破坏，时钟可能变得不同步。如果时钟不同步，这些协议的大多数可能受到攻击。如果发送者的时钟比接收者的时钟超前，M 能够截取从发送者传来的报文，当时间标记变成接收地当

前时间时，M 重放报文。这种攻击叫做隐瞒重放，其结果使人感到气愤。

这个协议是 Yahalom 协议的增强，下面是协议过程：

(1) A 把她的名字和随机数一起送给 B。

$$A, R_A \rightarrow B$$

(2) B 把 A 的名字连同她的随机数和一个时间标记 $T_B$ 一起，用他与 T 共享的密钥加密，并把加密的结果、B 的名字和一个新的随机数一起发给 T。

$$B, R_B, E_B(A, R_A, T_B) \rightarrow T$$

(3) T 产生随机会话密钥，然后产生两个报文，第一个报文由 B 的名字、A 的随机数 $R_A$、随机会话密钥 K 和时间标记 $T_B$ 组成，所有这些报文用他与 A 共享的密钥加密；第二个报文由 A 的名字、会话密钥 K 和时间标记组成 $T_B$，所有这些报文用他与 B 共享的密钥加密。他将这两个报文和 B 的随机数 $R_B$ 一起发给 A。

$$E_A(B, R_A, K, T_B), E_B(A, K, T_B), R_B \rightarrow A$$

(4) A 解出用她的密钥加密的报文，提出密钥 K，并确认 $R_A$ 与她在步骤(1)中的值相同。A 发给 B 两个消息，第一个是从 T 那里接收的用 B 的密钥加密的消息；第二个是用会话密钥 K 加密的 $R_B$。

$$E_B(A, K, T_B), E_K(R_B) \rightarrow B$$

(5) B 解出用他的密钥加密的消息，提出密钥 K，并确认 $T_B$ 和 $R_B$ 与它们在步骤(2)中有相同的值。

该协议过程如图 4-7 所示。

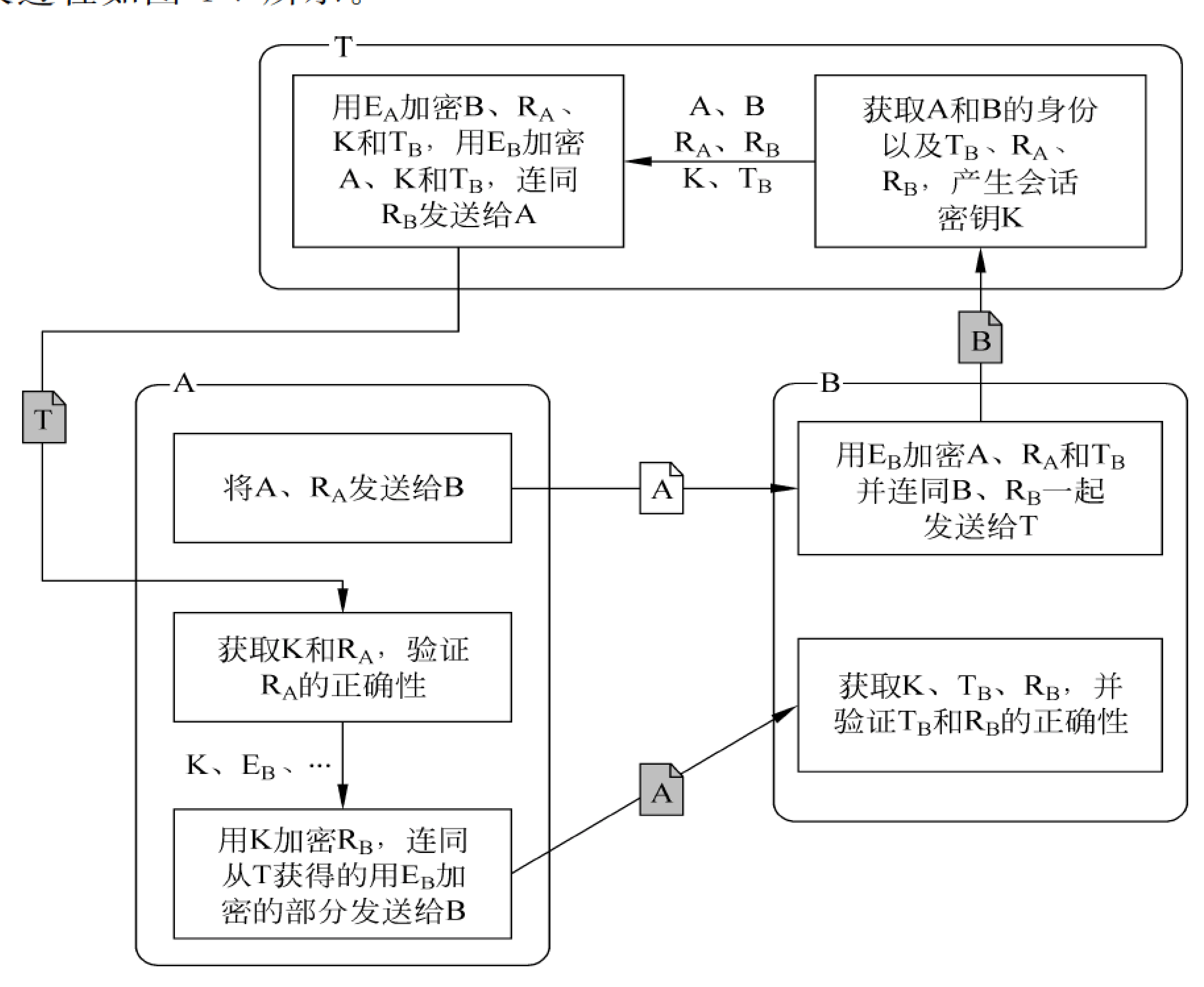



图 4-7 带时间标记和同步的协议

假设随机数和时间标记都匹配,A 和 B 就会相信互相的身份,并共享一个密钥。因为时间标记只是相对于 B 的时间,所以不需要同步时钟,B 只检查他自己产生的时间标记。

这个协议的好处是:在预先确定的时间内,A 能够用从 T 那里接收的消息与 B 作后续的认证。假设 A 和 B 完成了上面的协议和通信,然后终止连接,A 和 B 也不必依赖 T,就能够在 3 步之内重新认证。

(1) A 将 T 在步骤(3)发给她的信息和一个新的随机数发送给 B。

$$E_B(A,K,T_B),R'_A \rightarrow B$$

(2) B 发给 A 另一个新的随机数,并且 A 的新随机数用他们的会话密钥加密。

$$R'_B,E_K(R'_A) \rightarrow A$$

(3) A 用他们的会话密钥加密 B 的新随机数,并把它发送给 B。

$$E_K(R'_B) \rightarrow B$$

重新认证的协议过程如图 4-8 所示。

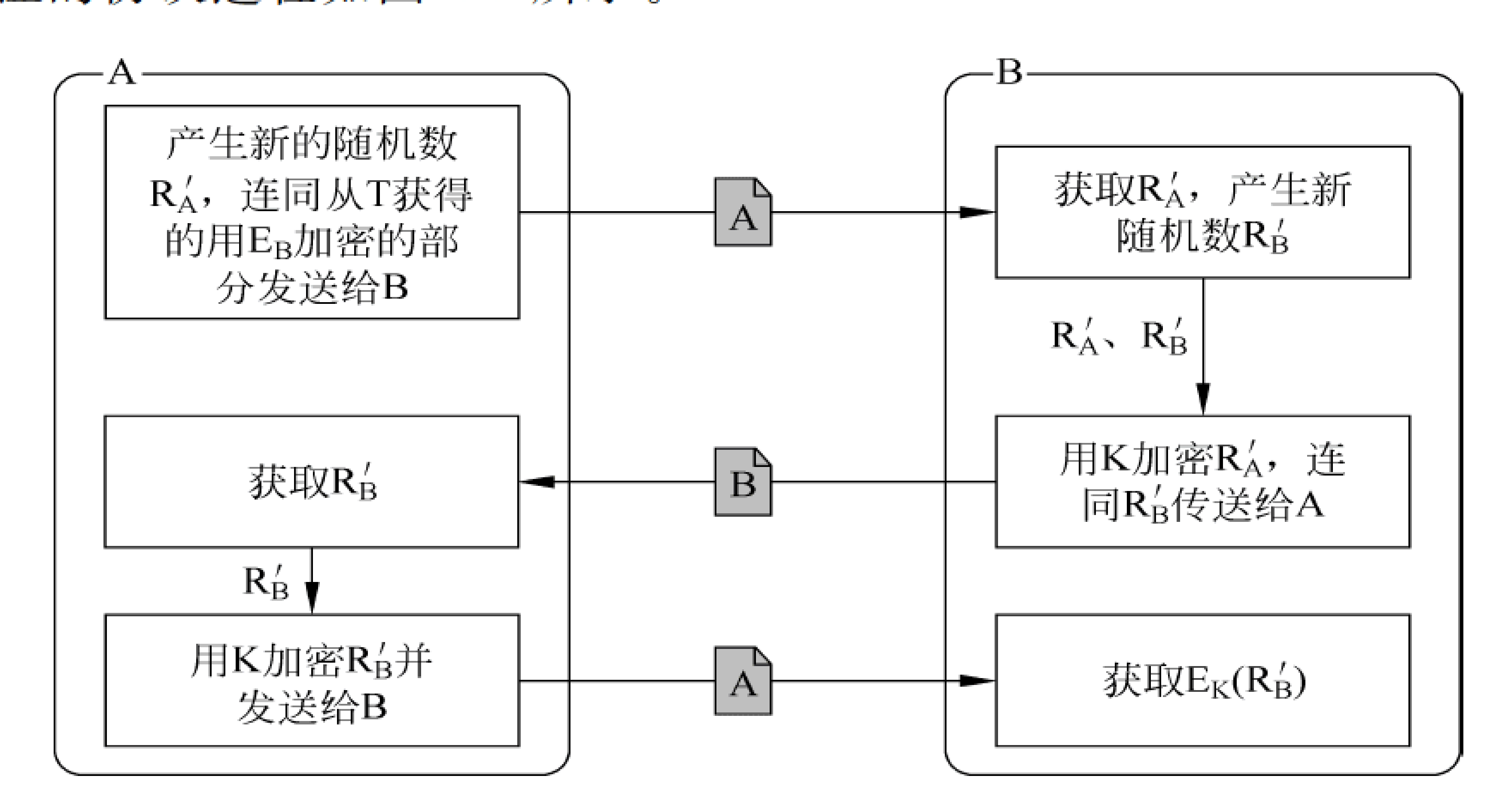



图 4-8 重新认证过程

这样,通过使用新的随机数就防止了重放攻击。

## 4.4.7 分布式认证安全协议

分布式认证安全协议(DASS)是由数字设备公司开发的,也提供相互认证和密钥交换。与前面的协议不同,DASS 同时使用了公开密钥和对称密码。A 和 B 每人有一个私钥,T 有他们公钥签名的副本。

(1) A 送一个消息给 T,这个消息由 B 的名字组成。

$$B \rightarrow T$$

(2) T 把 B 的公钥 $K_B$ 发给 A,用 T 的私钥 T 签名。签名消息包括 B 的名字。

$$S_T(B,K_B) \rightarrow A$$

(3) A 验证 T 的签名以确认她接收的密钥确实是 B 的公钥。A 产生一个随机会话密钥 K 和一个公钥/私钥密钥对 $K_P$,她用 K 加密时间标记,然后用她的私钥 $K_A$ 对密钥的寿命周期 L、A 的名字和 $K_P$ 签名。最后,她用 B 的公钥 $K_B$ 加密 K,并用 $K_P$ 签名。

A 将所有这些消息发给 B。

$$E_K(T_A), S_{KA}(L, A, K_P), S_{KP}(E_{KB}(K)) \rightarrow B$$

（4）B 发送一个消息给 T(这可能是另一个 T),它由 A 的名字组成。

$$A \rightarrow T$$

（5）T 把 A 的公钥 $K_A$ 和 A 的名字,用 T 的私钥 T 签名后发给 B。

$$S_T(A, K_A) \rightarrow B$$

（6）B 验证 T 的签名以确认他接收的密钥确实是 A 的公钥。然后他验证 A 的签名并恢复出 $K_P$。B 验证签名并用他的私钥恢复 K。然后解密 $T_A$ 以确认这是当前的消息。

（7）如果需要互相认证,B 用 K 加密一个新的时间标记,并把它发送给 A。

$$E_K(T_B) \rightarrow A$$

（8）A 用 K 解密 $T_B$ 以确认消息是当前的。

该协议过程如图 4-9 所示,图中假设 A 和 B 通信的为同一个 T。

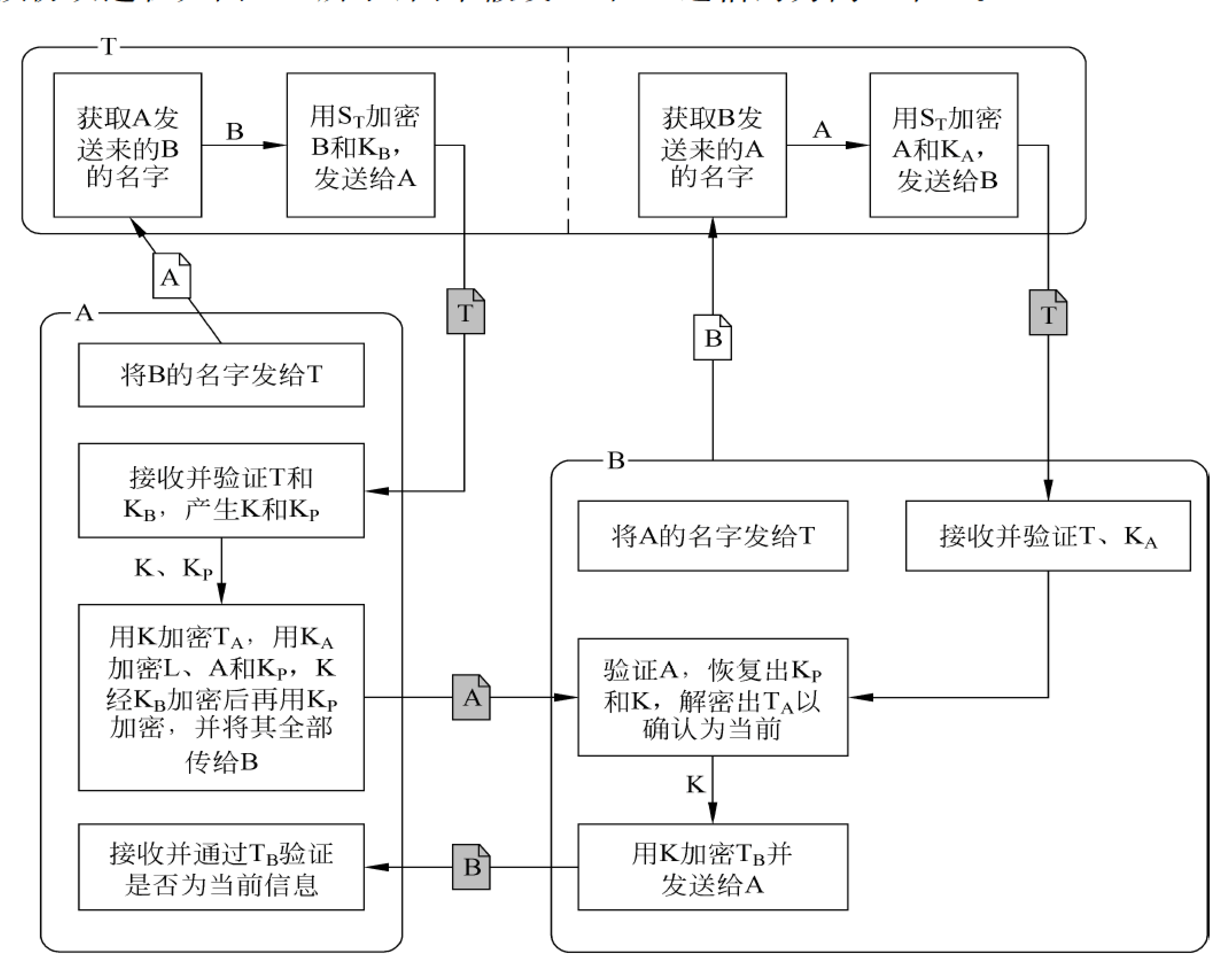



图 4-9　分布式认证安全协议

## 4.4.8　带 T 的公开密钥认证协议

这个协议也称为 Denning-Sacco 协议,它也使用公开密钥。T 保存每个人的公开密钥数据库。

(1) A 发送一个有关她和 B 的身份消息给 T。

$$A, B \rightarrow T$$

(2) T 把用 T 的私钥签名的 B 的公钥 $K_B$ 发给 A。T 也把用 T 的私钥签名的 A 自己的公钥 $K_A$ 发给 A。

$$S_T(B, K_B), S_T(A, K_A) \rightarrow A$$

(3) A 向 B 传送随机会话密钥、时间标记(用自己私钥签名并用 B 的公钥加密)和两个签了名的公开密钥。

$$E_B(S_A(K, T_A)), S_T(B, K_B), S_T(A, K_A) \rightarrow B$$

(4) B 用他的私钥解密 A 的消息,然后用 A 的公钥验证她的签名。B 检查以确认时间标记仍有效。

该协议过程如图 4-10 所示。

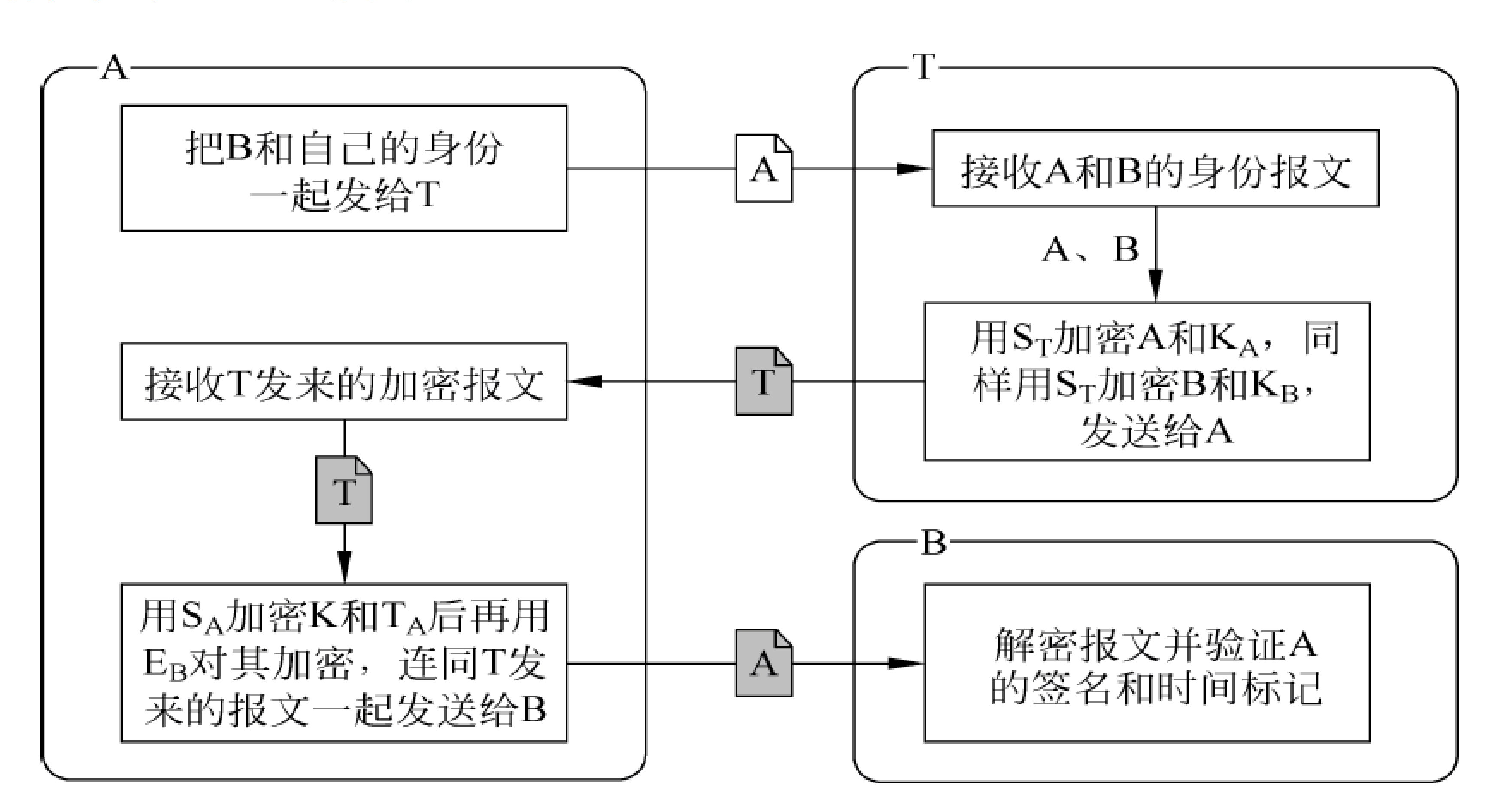



图 4-10　Denning-Sacco 协议

在这里,A 和 B 两人都有密钥 K,他们能够安全地通信。

这看起来是可行的,但实际上还存在问题。在和 A 一起完成协议后,B 可以伪装是 A。这个欺骗过程是:

(1) B 把他的名字和 C 的名字发给 T。

$$B, C \rightarrow T$$

(2) T 把 B 和 C 的已签名的公钥发给 B。

$$S_T(B, K_B), S_T(C, K_C) \rightarrow B$$

(3) B 将以前从 A 那里接收的会话密钥和时间标记的签名用 C 的公钥加密,并和 A 与 C 的证书一起发给 C。

$$E_C(S_A(K, T_A)), S_T(A, K_A), S_T(C, K_C) \rightarrow C$$

(4) C 用她的私钥解密 A 的消息,然后用 A 的公钥验证她的签名。C 检查以确认时间标记仍有效。

B 的上述欺骗过程如图 4-11 所示。

C 现在认为她正在与 A 交谈,但是 B 欺骗了她。事实上,在时间标记截止前,B 可以

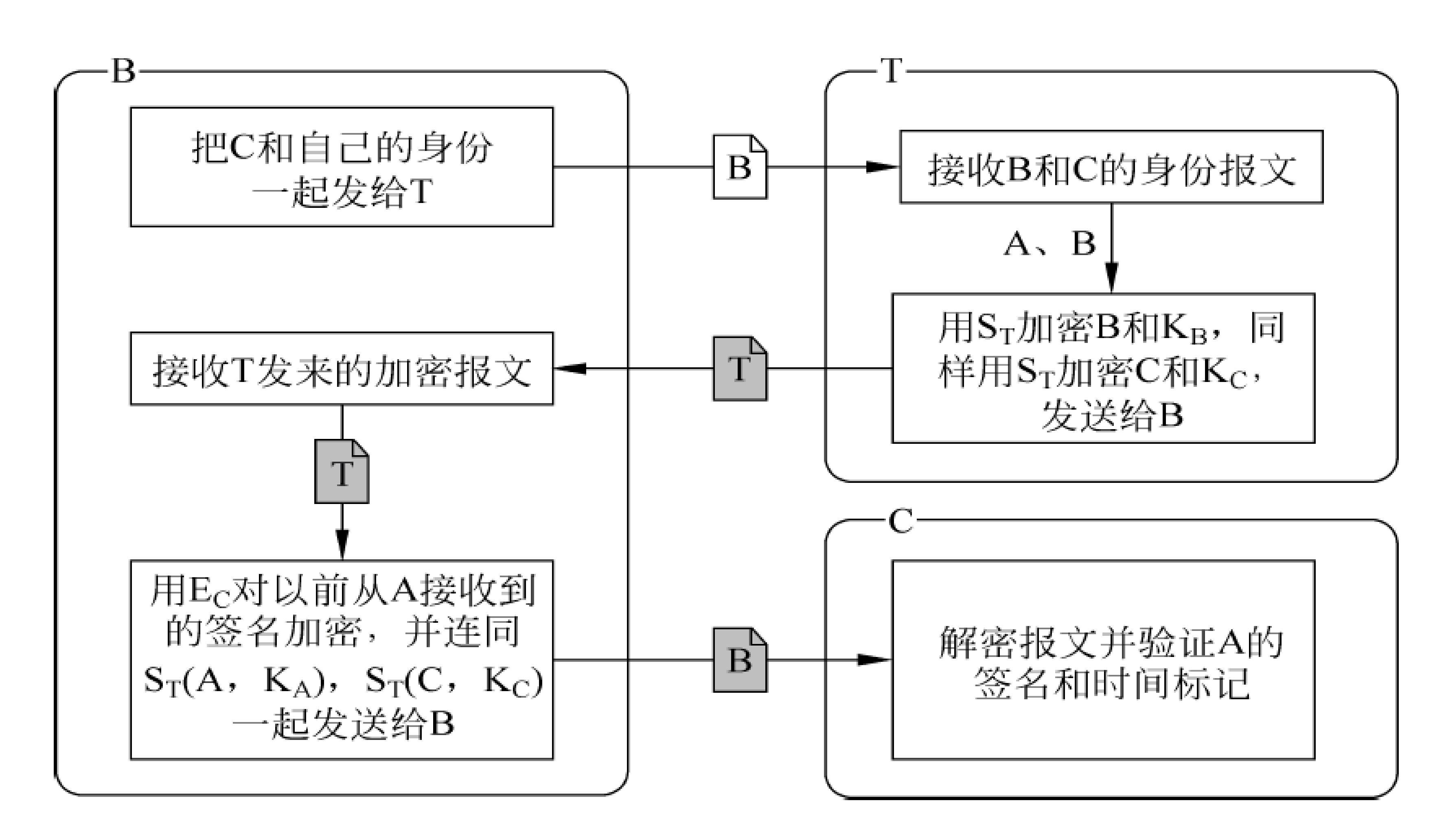



图 4-11　在 Denning-Sacco 中的欺骗过程

欺骗网上的任何人。

这个问题可以这样解决。在步骤(3)的加密消息内加上名字：

$$E_B(S_A(A,B,K,T_A)),S_T(A,K_A),S_T(B,K_B)$$

因为这一步清楚地表明是 A 和 B 之间在通信，所以 B 现在就不可能对 C 重放旧消息。

## 4.4.9　带 T 和随机数的公开密钥认证协议

这个协议也称为 Woo-Lam 协议，它也使用公开密钥：

(1) A 发送一个有关她和 B 的身份消息给 T。

$$A,B \rightarrow T$$

(2) T 用他的私钥 T 对 B 的公钥签名，然后把它发给 A。

$$S_T(K_B) \rightarrow A$$

(3) A 验证 T 的签名，然后把她的名字和一个随机数用 B 的公钥加密，并把它发给 B。

$$E_{K_B}(A,R_A) \rightarrow B$$

(4) B 把他的名字、A 的名字和用 T 的公钥 $K_T$ 加密的 A 的随机数一起发给 T。

$$A,B,E_{K_T}(R_A) \rightarrow T$$

(5) T 把用 T 的私钥签名的 A 的公钥 $K_A$ 发给 B，T 用 T 的私钥对所有 A 的随机数、随机会话密钥、A 的名字和 B 的名字签名，并用 B 的公钥加密，并把它也发给 B。

$$S_T(K_A),E_{KB}(S_T(R_A,K,A,B)) \rightarrow B$$

(6) B 验证 T 的签名。然后 B 将步骤(5)中 T 的消息的第二部分和一个新随机数一起用 A 的公钥加密，并将结果发给 A。

$$E_{KA}(S_T(R_A,K,A,B),R_B) \rightarrow A$$

(7) A 验证 T 的签名和她的随机数。然后她将第二个随机数用会话密钥 K 加密，并发给 B。

$$E_K(R_B) \rightarrow B$$

（8）B 解密他的随机数，并验证随机数有没有改变。

该协议过程如图 4-12 所示。

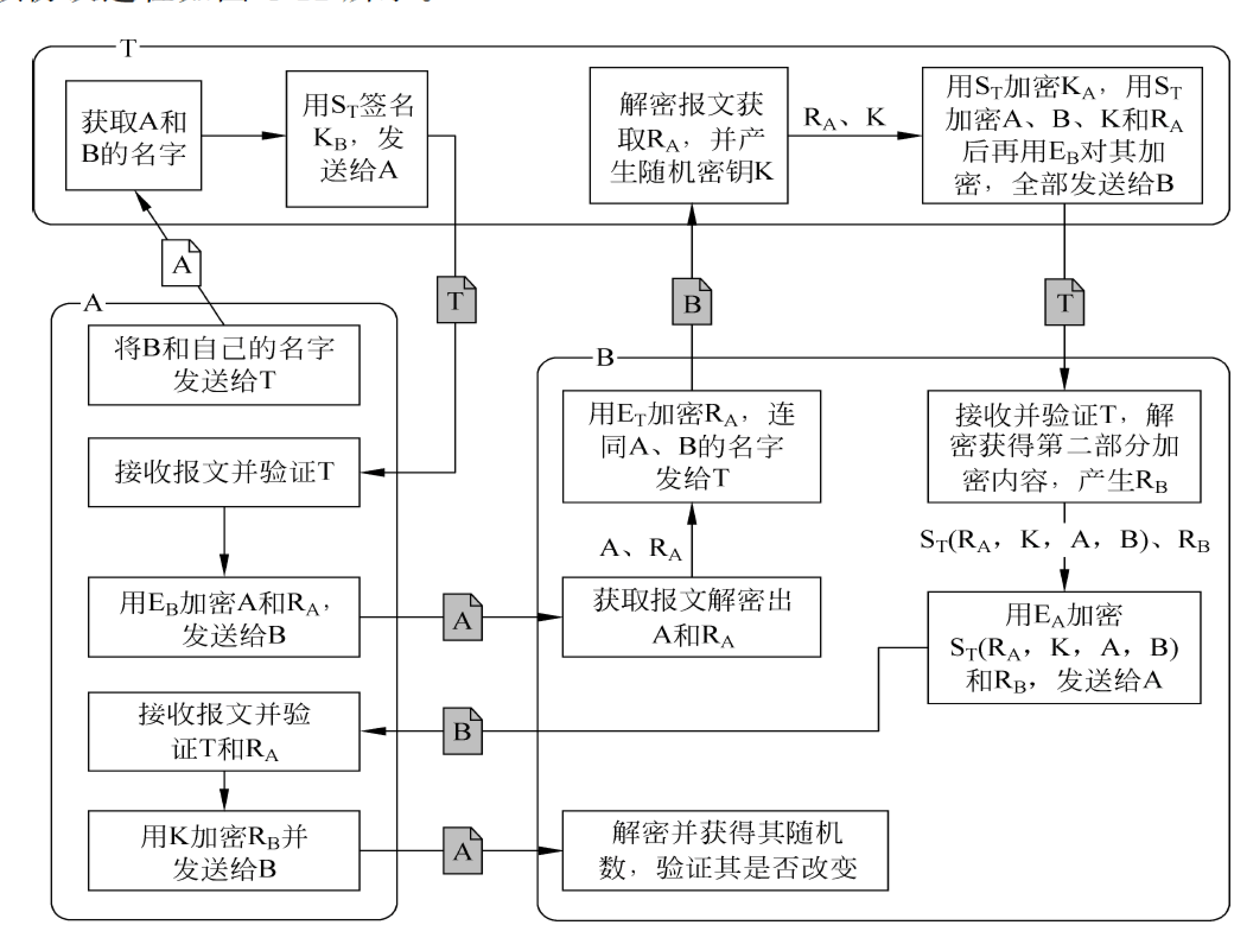



图 4-12　Woo-Lam 协议

## 4.4.10　其他协议

现有文献中有许多其他协议，如 X.509 协议、Krypto Knight 协议、加密密钥交换协议等。

## 4.4.11　学术上的教训

在前面的协议中，那些被破解的协议和没有被破解的协议都留下了沉痛的教训。

（1）因为设计者试图使设计过于精致，致使许多协议失败。在优化协议时，通过省去重要的部分：名字、随机数等来完成，但矫枉过正了。

（2）试图进行优化是一个可怕的陷阱，因为最终的成败依赖于所做的假设。例如，如果有识别的时间，就可以做许许多多没去做也做不到的事情。

（3）选择的协议依赖于底层的通信体系结构。因此，必须要考虑消息的大小和数量，还要考虑团体中的所有人是否需要互相交谈。

这些问题导致了协议分析的形式化方法的研究，参见 5.1 节。

## 4.5 多密钥公开密钥密码系统

一般地，公开密钥密码使用两个密钥，用一个密钥加密的报文能用另一个密钥解密。通常一个密钥是私有的，而另一个是公开的。假设 A 有一个密钥，B 有另一个密钥，那么 A 加密的报文，只有 B 能解密；反过来 B 加密的报文，只有 A 能读。

Colin Boyd 推广了这个概念。设想一种具有 3 个密钥 $K_A$，$K_B$，$K_C$ 的公开密钥密码系统。表 4-2 给出了密钥分配。

**表 4-2 3 个密钥分配方案**

| 人员 | 加密密钥分配 | 相应的解密密钥 |
|---|---|---|
| A | $K_A$ | $K_B$ 和 $K_C$ |
| B | $K_B$ | $K_A$ 和 $K_C$ |
| C | $K_C$ | $K_A$ 和 $K_B$ |
| D | $K_A$ 和 $K_B$ | $K_C$ |
| E | $K_B$ 和 $K_C$ | $K_A$ |
| F | $K_A$ 和 $K_C$ | $K_B$ |

在这个方案中，A 可以用 $K_A$ 加密报文，以便 E 用 $K_B$ 和 $K_C$ 解密；B 可以用 $K_B$ 对报文加密，以便 F 能读它；C 也可以用 $K_C$ 加密报文，以便 D 能读它；D 可以用 $K_A$ 加密报文以便 E 能读它；而由于有 $K_A$，F 也可以用 $K_A$ 加密报文；或者 D 同时用 $K_A$ 和 $K_B$ 加密报文，以便 C 能读它。

类似地，E 能够加密报文以便 A 和 D 或者 F 能读它。表 4-2 也归纳了所有可能的组合。

这可以推广到 n 个密钥，如果用密钥的某个子集来加密报文，那么就需要用其他密钥来解密此报文。

**广播报文**

假设有 100 个工人在野外作业，现在需要给他们当中的一些人发送报文，但预先并不知道是该向哪些人发。可以为每个人单独加密报文，或者为每种可能的组合都给出密钥。第一种选择需要增加很多通信量，第二种选择需要更多密钥。

采用多密钥密码方案就容易得多，将用 3 个工人：A、B 和 C。将 $K_A$ 和 $K_B$ 给 A，将 $K_B$ 和 $K_C$ 给 B，将 $K_C$ 和 $K_A$ 给 C。现在可以同想要通信的任何子集交谈。如果想发送一个报文只有 A 能读它，就用 $K_C$ 加密此报文。当 A 接收到此报文时，她先用 $K_A$，然后再用 $K_B$ 解密。如果想发送一个报文只有 B 能读它，就用 $K_A$ 加密；用 $K_B$ 加密时，只有 C 才能读它。如果发送一个报文使 A 和 B 都能读它，就用 $K_A$ 和 $K_C$ 对它加密等。

这可能没有什么激动人心的地方，但对于 100 个工人来说，它就非常管用了。

单独的报文表示每个工人共享一个密钥(总共 100 个密钥)和每个报文。每个可能子集的密钥表示共有 $2^{100}-2$ 个不同的密钥，另外的两个是：针对全部工人的报文和不对工人的报文。在这个方案里，只需要一个加密报文和 100 个不同的密钥就可以实现。

这个方案的缺陷是必须广播哪个工人的子集能够读报文。否则,每个工人将不得不试所有可能的密钥组合,以寻找正确的一组密钥。甚至只要意向接收者的名字也行得通。至少,要想实现简单,每个人实际上需要大量的密钥数据。

当然,还有其他用于报文广播的技术,其中有一些避免了前面的问题,这里不再细述。

# 4.6 秘密分割

设想,若某人发明了一种新的、特别好吃的食品配方,或者制作了一种调味品,现在重要的事情是必须保守秘密。只能告诉最信赖的雇员各种成分准确的调配比例,但如果他们中的一个背叛给对手方时,会出现怎样的情况呢? 很明显,秘密就会被泄露,不久之后,许多地方都将出售这种调味品。

为了避免这种情况的发生就需要对秘密进行分割。可以通过各种方法将消息分割成许多碎片。每一片本身并不代表什么,但把这些碎片放到一块,消息就会重现出来。如果消息是一个秘方,每一个雇员有一部分,那么只有他们放在一起才能作出这种调味料。如果任意一个雇员辞职带走一部分制法,这个信息本身是毫无用处的。

这就是所谓的秘密分割,它是指把一个消息分成 n 块,单独的每一块看起来并没有意义,但所有的块集合起来就能将原来的消息恢复出来。它包括简单的两方秘密分割和多方秘密分割两种情况。

在两个人之间分割一个消息是最简单的共享问题。下面是 T 把一消息分割给 A 和 B 的一个协议:

(1) T 产生一个随机比特串 R,和消息 M 一样长。

(2) T 用 R 异或 M 得到 S。

$$M \oplus R = S$$

(3) T 把 R 给 A,将 S 给 B。为了重构此消息,A 和 B 只需一起做一步。

(4) A 和 B 将他们的消息异或就可得到此消息。

$$R \oplus S = M$$

如果处理得当,这种技术是绝对安全的。因为每一部分本身是毫无价值的东西。

实际上,T 是用一次一密乱码本加密消息,并将密文给一个人,乱码本给另一个人。在 3.2.4 节中已详细讨论过一次一密乱码本,它们是具有完全保密性的,无论有多大计算能力都不能根据消息碎片之一的内容就确定出全部消息来。

把这种方案推广到多人也是容易的。为了在多个人中分割一个消息,将此消息与多个随机比特异或成混合物。

在下面的例子中,T 把信息划分成 4 部分:

(1) T 产生 3 个随机比特串 R、S、T,每个随机串与消息 M 一样长。

(2) T 用这 3 个随机串和 M 异或得到 U。

$$M \oplus R \oplus S \oplus T = U$$

(3) T 将 U 给 A,S 给 B,T 给 C,U 给 D。A、B 和 C、D 在一起可以重构此消息。

(4) A、B、C 和 D 一起计算。

$$R \oplus S \oplus T \oplus U = M$$

这是一个裁决协议，T 有绝对的权力，并且能够做他想做的任何事情。他可以把毫无意义的东西拿出来，并且申明这些是秘密的有效部分。在将秘密重构出来之前，没有人能够知道秘密所在。他可以分别交给 A、B、C 和 D 一部分，并且在之后告诉每一个人，只要 A、C 和 D 3 人就可以重构出此秘密，然后解雇 B。由于这是由 T 分配的秘密，这对于他恢复信息是没有问题的。

然而，这种协议存在一个问题：如果任何一部分丢失了，并且 T 又不在，就等于将消息丢掉了。如果 C 有调味料制法的一部分，他跑去为对手工作，并带走了他的那一部分，那么其他人就很不幸了，不可能重新产生这个秘方，即使 A、B、D 在一起也不行。C 的那一部分对消息来说和其他部分的组合一样重要。A、B 和 D 知道的仅是消息的长度，没有其他更多的信息了。的确如此，因为 R、S、T、U 和 M 都有同样的长度；见到它们中的任何一个都知道 C 的那一部分的长度。注意，M 不是通常单词意义的分割，它是用随机数异或的。

目前，秘密分割主要应用在数据库方面，通过对数据库信用卡号进行分割来提高其安全性。另外，在 2003 年由 RSA Security 公司推出的名为 Nightingale 的一项安全技术也体现和支持了秘密分割机制。

## 4.7 秘密共享

大家知道，对重要而敏感信息的保护主要还是采用加密手段来实现的，而加密的核心是密钥的保密问题。因此，密钥的管理直接影响信息的安全。

Blakley 和 Shamir 分别于 1979 年独立地提出了秘密共享的概念，并分别设计了具体的体制。秘密共享体制为密钥管理提供了一个非常有效的途径，在政治、经济、军事、外交中得到了广泛应用。

下面看两个秘密共享的例子。

**实例 1** 在一个银行里，每天都必须打开保险库，银行雇佣了 3 个出纳，但银行并不将密码委托给单个出纳。利用秘密共享体制就可以设计这样一个系统，在这个系统里，任何两个出纳合作都能打开保险库，但任何单独一个人就不能打开保险库。

**实例 2** 假设当前正在为核导弹安装发射程序。需要保证一个疯子不能够启动发射。也许要保证两个疯子不能启动发射。而在允许发射前，可以认为，5 个官员中至少有 3 个是疯子。这是一个秘密共享问题。做一个机械发射控制器，给 5 个官员每人一把钥匙，并且在允许他们起爆时，要求至少 3 个官员的钥匙插入合适的钥匙孔中。如果确实担心，可以使这些钥匙孔分隔很远，并要求官员们同时将钥匙插入。因为不愿一个官员偷窃另两把钥匙，使他能够毁坏城市。

除此以外，在重要场所的通行，遗嘱的生效等必须有两人或多人同时参加才能生效，也需要将秘密分给多人保管，即都要采用秘密共享体制。秘密共享体制不仅在密钥管理方面大有可为，而且在密钥的产生、分配方面也有一定的应用。

### 4.7.1　秘密共享的基本思想

秘密共享的一般思想是：为了解密一种秘密，需要多方协作才能完成。某个人只有自己的密钥还不够，还需要另外的一些帮助才能获取整个秘密。结果是：可以设计复杂的方案，方案明确指定哪些人必须相互协作，才可以解密特定消息。例如，可以指定一个“菜单”方法，在这种方法里，需要两个 A 栏中的人、3 个 B 栏中的人和一个 C 栏中的人来解密一条消息。采用的方案也可以有更复杂的相关性。例如，如果 A 使用了自己的密钥，那么还需要 B 的帮助；如果 C 使用了自己的密钥，还需要 D 的帮助（只有一种组合有效）。秘密共享最简单的例子是 4.6 节所述的秘密分割。

在上面的例子中，可以授权将军和两个上校发射导弹，但如果将军正在打高尔夫球，那么就要求启动发射时需要 5 位上校。这样，可以制造一种发射控制器，该控制器需要 5 把钥匙。给将军 3 把钥匙，给每位上校一把钥匙。实施的时候，将军和任何两位上校一起就能发射导弹。5 个上校一起也能，但将军和一位上校就不能，4 位上校也不行。

一种成为门限方案的共享方案，在数学上可以满足这个要求，甚至提供更多的条件。至少，对于任何消息（秘密的秘方，发射代码，洗衣价目表），可以把它分成 n 部分，每部分叫做它的“影子”或共享，使得它们中的任何 m 部分能够用来重构消息，更准确地说，这叫做秘密共享的（m，n）门限方案，或称拉格朗日插值多项式方案。该方案的一般描述如下。

假定需要在 n 个人中共享秘密 M，使得他们中任意 m 个人通过相互协作可以获取秘密。

（1）生成比 M 大的随机质数 p。

（2）生成（n－1）个随机整数 R{1}、R{2}，…，R{n－1}，每一个都比 p 小。

（3）使用下列式子将 F(x)定义成有限域上的多项式：

$$F(x)=(R\{1\}*x^{n}+R\{2\}*x^{(n-1)}+\cdots+R\{n-1\}*x+M) \bmod p \tag{4-1}$$

（4）通过定义：

$$k\{i\}=F(x\{i\}) \tag{4-2}$$

生成 F 的 m“影子”，这里，每个 x{i}都不同（对于 x，使用连续整数值[1，2，3，…]是很好的选择）。

（5）将[p，x{i}，k{i}]交给 m 个秘密共享者的每一位，i 对应每个共享者的号码（这一枚举是任意的）。

（6）销毁 R{1}、R{2}、…、R{n－1}。

（7）销毁或隐藏 M。

已知所提供的信息，每位秘密共享者可以写出一个线性方程。例如，作为共享者 1 可以构造方程：

$$k\{1\}=(C\{1\}*x\{1\}^{n}+C\{2\}*x\{1\}^{(n-1)}+\cdots+C\{n-1\}*x\{1\}+M) \bmod p \tag{4-3}$$

由于这些线性方程有 n 个未知数（C{1}，…，C{n－1}和 M），因此它需要 n 个具有相同未知数的方程来求解方程系统，进而解出 M（也可以解出 C{i}，但是一旦获取了 M，就对它们不感兴趣了）。

由于 F 的系数是随机选择的，所以少于 n 个秘密共享者的协作（即使具有无限的计

算能力)也不能解出 M。没有第 n 个共享者的参与,任何可能的 M(长度小于 p)都和任何其他 M 一样与(少于 n 个)方程相容。

再拿(3,4)门限方案来说,T 可以将他的秘密配方分给 A、B、C 和 D,这样把他们中的任意 3 个“影子”放在一起就能重构消息。如果 C 正在度假,那么 A、B 和 D 可以联合起来重构消息。如果 B 被汽车撞了,那么 A、C 和 D 可以联合起来重构消息。然而,如果 C 正在度假期间,B 被汽车撞了,A 和 D 就不可能重构消息了。

普通的门限方案远比上面所说的更通用。任何共享方案都能用它建造模型。

一个人可以把消息分给其所在大楼中的人,以便今后重构它来使用。这人需要一楼的 7 个人和二楼的 5 个人,就能恢复此消息。如果有从三楼来的人,在这种情况下,仅需要从三楼来的人和从一楼来的 3 个人及从二楼来的 2 个人;如果有从四楼来的人,在这种情况下,仅需要从四楼来的人和从三楼来的 1 个人,或从四楼来的人和从一楼来的 2 个人及从二楼来的 1 个人等。

秘密共享在具体的研究和实践中已得到了广泛应用。在自组织网络当中,由于该网络本身的灵活性给其安全方面提出了巨大挑战,而密钥的管理是其安全需求的首要环节,那么利用可公开验证的秘密共享体制而产生的分布式密钥管理办法就为提高自组织网络的安全性提供了一条有效途径。另外,秘密共享更广义上的应用——多秘密共享机制,在企业管理、遗产继承等方面也有大量应用。

## 4.7.2 基于秘密共享的协议

### 4.7.2.1 Shamir 门限方案

Shamir 的方法是将一个含有秘密的数据 $D$(例如,密码系统中的密钥,保险柜的号码组合等)分成 $n$ 块 $D_1,\cdots,D_n$,并满足下面的要求:

(1) 知道任意 $k$ 个或更多的 $D_j$,就能够有效地计算出 $D$。

(2) 知道任意 $k-1$ 个或更少的 $D_j$,由于信息不够,不可能有效地计算出 $D$。Shamir 称这种方法为$(k,n)$门限方案。

显然,这是对密钥保管问题的一种较好的解决方法。例如,在$(k,n)$门限方案中令 $n=2k-1$,则在 $n$ 个小块中即使有 $\lfloor n/2 \rfloor=k-1$ 个小块被破坏了,仍然可以有效地恢复原来的密钥。相反,密码分析员即使能够获得剩下的 $k$ 个小块中的 $\lfloor n/2 \rfloor=k-1$ 块,也无法有效地推导出原有的密钥。

Shamir 的门限方案是基于拉格朗日的插值多项式。在二维平面上给出 $k$ 个点 $(x_1,y_1),\cdots,(x_k,y_k)$,其中 $x_i$ 各不相同,则有一个且仅有一个 $k-1$ 次多项式 $q(x)$满足

$$q(x_i)=y_i,1\leqslant i\leqslant k \tag{4-4}$$

不失一般性,可以假定数据 $D$ 是一个数,为了把 $D$ 分成小块 $D_j$,选取一个随机的 $k-1$ 次多项式

$$q(x)=a_0+a_1x+\cdots+a_{k-1}x^{k-1} \tag{4-5}$$

其中,$a_0=D$,并计算出

$$D_1=q(1),\cdots,D_n=q(n) \tag{4-6}$$

给定上述 $D_j$ 值中的任意 $k$ 个,可以通过插值法求出 $q(x)$的系数,因此可以计算出 $D=q(0)$。相反,如果仅仅知道这些 $D_j$ 值中的 $k-1$ 个,则由于信息不够而无法确定 $q(x)$,所以不能求出 $D$ 值。

以上是 Shamir 门限方案的概述,下面对若干细节予以说明。

(1) 这里进行的是模数运算,而不是实数运算。因为插值定理在任何多项式环 F[x]中(其中多项式系数的集合 F 是一个域)都成立,且当 $p$ 是素数时,$Z_p$ 是一个域。所以给定整数 $D$ 后,选取一个大于 $D$ 和 $n$ 的素数 $p$,此后所进行的计算,都是模 $p$ 下的运算。

(2) $q(x)$中的系数 $a_1,\cdots,a_{k-1}$ 是从$[0,p-1]$中整数的均匀分布中随机地选取的。$D_1,\cdots,D_k$ 的计算,都是模 $p$ 下的运算。

(3) 当给定 $k$ 个小块 $D_{j_1},D_{j_2},\cdots,D_{j_k}$ 时,可以根据拉格朗日多项式,重新构造 $q(x)$:

$$q(x)=\sum_{i=1}^{k}D_{j_i}\prod_{\substack{s=1\\ s\neq i}}^{k}\frac{x-x_{j_s}}{x_{j_i}-x_{j_s}}\bmod p \tag{4-7}$$

因为进行的是模 $p$ 下的运算,所以式(4-7)中的除法是通过求出模 $p$ 的逆之后再进行乘法运算实现的。

(4) 密码分析员得到 $k-1$ 块 $D_{j_1},\cdots,D_{j_{k-1}}$ 的情形,任何$[0,p-1]$中的整数 $D$ 都有可能成为真正 $D$ 的候选值。对每一个 $D$,密码分析员能构造且仅能构造一个 $k-1$ 次多项式 $q(x)$,满足

$$q'(0)=D' \tag{4-8}$$

及

$$q'(j_1)=D_{j_1},\cdots,q'(j_{k-1})=D_{j_{k-1}} \tag{4-9}$$

由构造方法可知,这 $p$ 个可能的多项式作为 $q(x)$的概率都是相等的,密码分析员得不到能够帮助其推导出 $q(x)$的任何信息。

因此,根据上面的分析可以得出如下的结论:Shamir 的门限方案满足本节开始时所阐述的两项要求。

下面举一个具体的例子。

**例**　令 $k=3,n=5,p=19,D=11$。在[0,18]中随机地选取 $q(x)$的系数 $a_1=2,a_2=7$,因此

$$q(x)=(7x^2+2x+11)\bmod 19 \tag{4-10}$$

分别计算出

$$D_1=q(1)=(7+2+11)\bmod 19=20\bmod 19=1 \tag{4-11}$$

$$D_2=q(2)=(28+4+11)\bmod 19=43\bmod 19=5 \tag{4-12}$$

$$D_3=q(3)=(63+6+11)\bmod 19=80\bmod 19=4 \tag{4-13}$$

$$D_4=q(4)=(112+8+11)\bmod 19=131\bmod 19=17 \tag{4-14}$$

$$D_5=q(5)=(175+10+11)\bmod 19=196\bmod 19=6 \tag{4-15}$$

确定其中 3 个小块 $D_2=5,D_3=4$ 和 $D_5=6$,就可以通过拉格朗日插值多项式求出 $q(x)$。从而有

$$5\,\frac{(x-3)(x-5)}{(2-3)(2-5)}=5\,\frac{(x-3)(x-5)}{(-1)(-3)}$$
$$=5\,\frac{(x-3)(x-5)}{3}$$
$$=5\cdot\mathrm{inv}(3,19)\cdot(x-3)(x-5)$$
$$=65(x-3)(x-5) \tag{4-16}$$

其中，$\mathrm{inv}(3,19)=13$ 表示 $3\cdot 13\equiv 1(\mathrm{mod}\ 19)$。此外，还有

$$4\,\frac{(x-2)(x-5)}{(3-2)(3-5)}=4\,\frac{(x-2)(x-5)}{(1)(-2)}$$
$$=4\,\frac{(x-2)(x-5)}{-2}$$
$$=4\cdot\mathrm{inv}(-2,19)\cdot(x-2)(x-5)$$
$$=4\cdot\mathrm{inv}(17,19)\cdot(x-2)(x-5)$$
$$=36(x-2)(x-5) \tag{4-17}$$

及

$$6\,\frac{(x-2)(x-3)}{(5-2)(5-3)}=6\,\frac{(x-2)(x-3)}{(3)(2)}$$
$$=6\,\frac{(x-2)(x-3)}{6}$$
$$=6\cdot\mathrm{inv}(6,19)\cdot(x-2)(x-3)$$
$$=6\cdot 16\cdot(x-2)(x-5)$$
$$=96(x-2)(x-3) \tag{4-18}$$

所以

$$q(x)=[65(x-3)(x-5)+36(x-2)(x-5)+96(x-2)(x-3)]\mathrm{mod}\ 19$$
$$=[8(x-3)(x-5)+17(x-2)(x-5)+(x-2)(x-3)]\mathrm{mod}\ 19$$
$$=(26x^2-188x+296)\mathrm{mod}\ 19$$
$$=7x^2+2x+11 \tag{4-19}$$

因此，只要知道 $D_2=5$，$D_3=4$ 和 $D_5=6$，就可以求出

$$D=q(0)=11 \tag{4-20}$$

如果尝试任意其他 3 个 $D_j$ 值，也可以得到相同的结果。

#### 4.7.2.2 Asmuth-Bloom 门限方案

1980 年，Asmuth 和 Bloom 提出了一种基于中国剩余定理的$(k,n)$门限方案。这里，各小块 $D_i$ 与一个和 $D$ 相关的数 $D'$ 同余。在俩人的方案中，令

$$\{p,d_1,d_2,\cdots,d_n\}$$

为满足下述条件的整数：

(1) $p>D$

(2) $d_1<d_2<\cdots<d_n$

(3) $\gcd(p,d_i)=1,1\leqslant i\leqslant n$

(4) $\gcd(d_i, d_j)=1, i\neq j, 1\leqslant i, j\leqslant n$

(5) $d_1 d_2 \cdots d_k > p d_{n-k+2} d_{n-k+3} \cdots d_n$

条件(3)和(4)说明,这组整数$\{p, d_1, d_2, \cdots, d_n\}$是两两互素的;条件(5)说明,$k$ 个最小的 $d_i$ 的乘积大于 $p$ 和 $k-1$ 个最大的 $d_i$ 的乘积。令 $m=d_1 d_2 \cdots d_k$ 是 $k$ 个最小的 $d_i$ 的乘积,则 $m/p$ 大于任何 $k-1$ 个 $d_i$ 的乘积。在$[0,(m/p)-1]$的范围内随机地选取一个 $r$,计算出 $D'=D+rp$。由 $r$ 的选取方法和条件(1)可知,$D'$一定在$[0, m-1]$的范围之内。最后,如下计算出 $n$ 个 $D_i$ 块:

$$D_i = D' \bmod d_i, 1 \leqslant i \leqslant n \tag{4-21}$$

只要知道上述 $D_i$ 块中的任意 $k$ 个,例如,$D_{i_1}, \cdots, D_{i_k}$,就可以应用中国剩余定理求出 $D'$:

$$y \equiv D' (\bmod\ m_1) \tag{4-22}$$

其中 $m_1=d_{i_1} d_{i_2} \cdots d_{i_k}$。因为 $m_1 \geqslant m$,所以上述的 $D'$是唯一确定的。求出 $D'$后,就不难算出

$$D = D' - rp$$

相反,如果仅仅知道 $k-1$ 块 $D_i$,即知道 $D_{i_1}, \cdots, D_{i_{k-1}}$,则只能应用中国剩余定理求出满足下列同余式

$$y \equiv D_{i_j} (\bmod\ d_{i_j}), 1 \leqslant j \leqslant k-1 \tag{4-23}$$

的

$$y \equiv D'' (\bmod\ m_2) \tag{4-24}$$

其中 $m_2=d_{i_1} \cdots d_{i_{k-1}}$。因为 $m/m_2 > p$,且有 $\gcd(m_2, p)=1$,所以在$[0, m]$中与 $D''$模 $m_2$ 同余的数在所有的模 $p$ 的同余类中均匀地分布,即使没有足够的信息能够确定出 $D'$。下面用一个具体的例子说明上述同余类$(k, n)$门限方案。

**例**　令 $k=2, n=3, D=4, p=7, d_1=9, d_2=11, d_3=13$。

因为 $m=d_1 d_2=9 \cdot 11=99>91=7 \cdot 13=p \cdot d_3$,所以满足同余类方案的要求。可以在

$$[0, 99/7-1] = [0, 13]$$

中随机地选取一个 $r=10$,因此

$$D' = D + rp = 4 + 10 \cdot 7 = 74$$

分别计算出

$$D_1 = 74 \bmod 9 = 2$$

$$D_2 = 74 \bmod 11 = 8$$

$$D_3 = 74 \bmod 13 = 9$$

由上述计算可知

$$y \equiv D' (\bmod\ 9 \cdot 11 \cdot 13)$$

是建立同余式

$$\begin{cases} y \equiv 2 (\bmod\ 19) \\ y \equiv 8 (\bmod\ 11) \\ y \equiv 9 (\bmod\ 13) \end{cases}$$

的解,其中 $D'=74$。

如果知道上面 $D_i$ 中的任意两个,就可以计算出 $D$。假如,已知 $D_1=2$ 和 $D_2=8$,则有

$$m_1 = d_1 d_2 = 9 \cdot 11 = 99$$

为了引用中国剩余定理,首先要求出

$$y_1 = \mathrm{inv}(m_1/d_1, d_1) = \mathrm{inv}(11,9) = 5$$

和

$$y_2 = \mathrm{inv}(m_1/d_2, d_1) = \mathrm{inv}(9,11) = 5$$

其中 inv(a,b)=c 表示 a·c≡1(mod b)。因此

$$\begin{aligned} D' &= \left[\left(\frac{m_1}{d_2}\right)y_1 D_1 + \left(\frac{m_2}{d_2}\right)y_2 D_2\right] \bmod m_1 \\ &= [11 \cdot 5 \cdot 2 + 9 \cdot 5 \cdot 8] \bmod 99 \\ &= [110 + 360] \bmod 99 \\ &= 74 \end{aligned}$$

最后,计算出

$$D = D' - rp = 74 - 10 \cdot 7 = 4$$

Asmuth 和 Bloom 设计了一个有效的重新构造 $D$ 的算法,其运算量是 $O(k)$,存储量 $O(n)$。因此,他们的方案更有效于前述的 Shamir 多项式方案,回忆 Shamir 的门限方案所需要的工作量是 $O(k\log^2 k)$。当然,在 $D$ 值不大的情况下,这两个方案区别不大。

最后阐述一种有意义的观点。门限方案通过不供给任何 $k-1$ 个 $D_j$ 块足够的信息重新构造 $D$ 的方法,达到了无条件的保密性。那么,如何解释在第 2 章曾经讲过的“一次一密系统是唯一的无条件保密的密码系统”这一事实呢? Blakley 指出,可以把一次一密系统视为一个保护信息 $M$ 的 $k=2$ 的门限方案。这里,发送者和接收者各有一个密钥 $M_1$ 的复制,将 $M_1$ 看成是一块 $D_j$,即 $D_1=M_1$。第 2 块 $D_j$ 是由发送者构造并传送给接收者的密文 $M_2=M\oplus M_1$,即 $D_2=M_2$。为了重新构造出 $M$,必须同时具备 $M_1$ 和 $M_2$,缺一不可。这种将一次一密系统归类为密钥门限方案的观点的意义在于,从另一个角度解释了只有一次一密系统,而不是其他密码系统是理论上不可破的密码系统的原因。

### 4.7.3 秘密共享的例子

**例 1** 有骗子的秘密共享

有许多方法可欺骗门限方案,下面是几种可能的情景。

**情景 1** 来自内部的人员不配合:上校 A、B 和 C 在某个隔离的地下掩体中。一天,他们从总统那里得到密码消息:“发射导弹,根除这个国家的神经网络”。于是要求 A、B 和 C 出示他们的“影子”,一起发射导弹。但 C 却输入一个随机数,因为她实际上是和平主义者,并不想发射导弹。结果,由于 C 没有输入正确的“影子”,因此他们恢复出来的秘密是错误的,导弹还是停放在发射井中,不能发射。更麻烦的是,没有人知道为什么会这样。即使 A 和 B 一起努力,他们也不能证明 C 的“影子”是无效的。

**情景 2** 来自外部的人员窃取内部人员的“影子”:上校 A 和 B 与 M 正坐在掩体中。M 假装也是上校,他并没有合法的“影子”,而其他人都没有办法识破。同样要求发射导

弹的消息从总统那里来了，并且每人都出示了他们的“影子”，“哈，哈！”M 说，“我伪造了从总统那里来的消息，现在我知道你们两人的‘影子’了”。

**情景 3**　来自外部的人员窃取秘密：上校 A、B 和 C 与 M 一起坐在掩体中，M 还是伪装的，没有合法的影子。同样要求发射导弹的消息从总统那里来了，并且每人都出示了他们自己的“影子”，而 M 在看到他们 3 人的“影子”之后，才出示自己的“影子”。由于要重构这个秘密只需要 3 个“影子”，因此 M 可以很快地产生一个有效的“影子”并出示。这样，M 不仅知道了秘密，而且还没有人知道他并不是这个秘密共享方案的一部分这个事实。

**例 2**　没有 T 的秘密共享

假设在 5 个官员中必须要有 3 个人同时插入他们的钥匙才能打开银行的金库。没有人知道整个秘密，也没有裁判者 T 来把秘密分成 5 部分，而是使用一种 5 个官员可以恢复秘密的协议。通过这个协议，每人分得一部分秘密，而官员们在重构秘密之前，无人知道这个秘密。

在本书中不准备讨论这些协议。

**例 3**　不暴露共享的共享秘密

上述方案有一个问题，就是当每个人聚到一起重构秘密时，就可能暴露他们自己的秘密。

其实可以采用这种方法避免出现这种情况：如果最终的共享秘密是私钥（例如数字签名），那么 n 个共享者中的每一个都可以完成文件的一部分签名。在第 n 部分签名后，文件已经用共享的私钥签名，并且共享者中没有人了解任何其他人的秘密。

这个例子要说明的是，秘密能够重用是关键，并且不必用可信的处理器去处理它。

**例 4**　可验证的秘密共享

假设 T 给 A、B、C 和 D 每人分配了一部分秘密。他们中的任何人想知道他们是否持有有效秘密部分，唯一的办法就是尝试着去重构秘密。假设 T 发给 B 一个假的共享秘密，或者由于通信错误，B 偶然接收到一个坏的共享秘密。因此要求可验证的秘密共享方案能够允许他们中的每个人分别验证自己持有一个有效的共享秘密，而不用重构这个秘密。

**例 5**　带预防的秘密共享方案

假设一个秘密被分给 50 个人，只要其中的任何 10 个人在一起，就可以重构这个秘密。一般情况下，这个要求是容易满足的。但是，如果增加约束条件，要 20 人在一起才能恢复秘密，同时还要防止其他人重构秘密时，我们能实现这种秘密共享方案吗？是不是有多少人共享秘密都没有问题？

从数学上看，基本思想是每个人得到两个共享秘密的可能：一个“是”和一个“否”。当重构秘密时，每个人提交他们的一个共享。他们提交的实际共享依赖于他们是否希望重构秘密。如果有 m 或更多个“是”共享，和少于 n 个“否”共享，那么秘密就能够被重构，否则，不能重构。

当然，如果没有“否”共享的人，没有任何事情能防止足够数量的“是”共享的人钻牛角尖，去重构秘密。但是在每个人提交他们的共享进入中心计算机的情况下，这个方案

可行。

**例 6** 带除名的秘密共享

假设某人正在安装秘密共享系统，现在他想解雇一名共享者。虽然可以安装没有那个人的新方案系统，但很费时。现在要设计一套方案，在原有秘密共享系统的基础上，加上除名的功能。

有多种方法处理这个系统，一旦有一个参与者变成不可信时，就可以立即启用新的共享方案。

# 4.8 数据库的密码保护

## 4.8.1 数据库安全的重要性

数据库是当今信息社会中数据存储和处理的核心，其安全性对于整个信息安全极为重要。

首先，数据库安全对于保护组织的信息资产非常重要。组织中绝大部分信息资产都是保存在数据库中的，拥有这些信息资产的组织必须保证这些信息不被非授权的形式访问。

其次，保护数据库系统所在网络系统和操作系统非常重要，但仅仅如此远不足以保证数据库系统的安全。很多有经验的安全专业人士有一种常见的误解——一旦评估和消除了服务器上的网络服务和操作系统的脆弱性，该服务器上所有应用就都是安全的了。实际上，现代的数据库系统有很多特征可以被误用或利用来损害系统中的数据安全。

再次，数据库安全的不足不仅会损害数据库本身，而且还会影响到操作系统和整个网络基础设施的安全。例如，很多现代数据库都有内置的扩展存储过程，如果不加控制，攻击者就可以利用它来访问系统中的资源。

最后，数据库是电子商务、电子政务、ERP 等关键应用系统的基础，它的安全也是这些应用系统的基础。

总之，数据库系统在给人们带来了方便和效率的同时，也带来了对安全方面更高的要求。因此，数据库的安全问题十分重要，必须给予高度重视。

## 4.8.2 数据库的安全问题

为了使数据库的安全性得到保证和不断加强，首先需要知道数据库所存在的安全问题。在本节中，对数据库的安全威胁和安全需求两个方面进行说明。

通过图 4-13 可以看出，数据库面临着严重的安全威胁。

具体来讲，主要有以下因素：

（1）数据输入或处理中的错误。例如，准备输入的数据在输入前已被修改，有的机密数据在输入到计算机之前已被公开，在数据处理操作中的误操作等，均会使数据出错。

（2）硬件故障引起的信息破坏或丢失。例如，操作系统设计上的缺陷，缺少存取控制

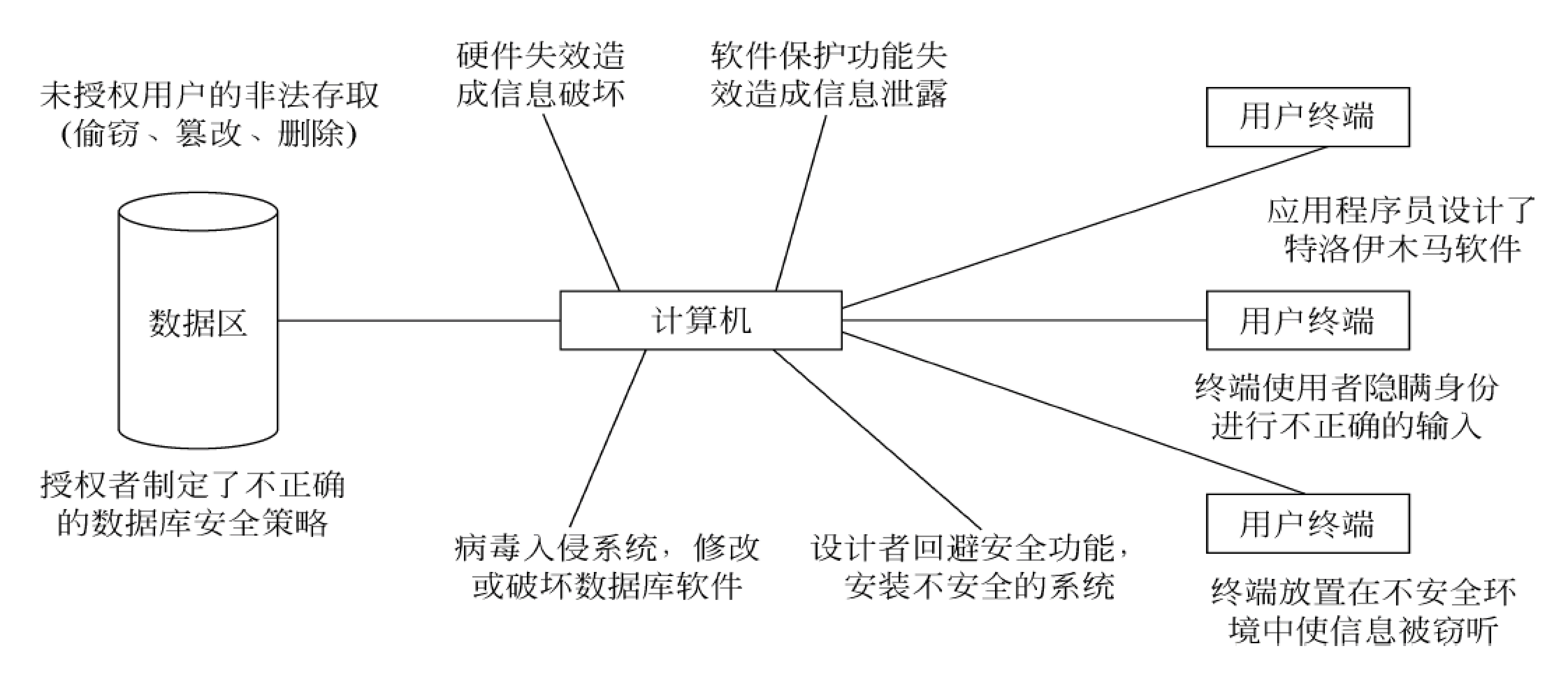



图 4-13　数据库面临的安全威胁

或破坏了存取控制机制，造成信息泄露。

（3）软件保护功能失效造成信息泄露。例如，操作系统设计上的缺陷，缺少存取控制机制或破坏了存取控制机制，造成信息泄露。

（4）非授权用户的非法存取或篡改数据。例如，数据库管理人员对数据的使用权限不进行严格的管理，对哪些用户有数据访问权、哪些用户有数据修改更新权等缺乏严格的检查控制措施；对用户在计算机上的活动没有进行监督检查，致使非授权用户非法存取，合法用户对数据进行篡改。

（5）授权者指定不正确、不安全的防护策略。

（6）操作者复制和泄露机密、敏感数据资料。

（7）系统设计者回避安全功能，安装不安全的系统。

（8）应用程序员设计、安装了特洛伊木马软件。

（9）终端放置在不安全的环境中而被窃听。

（10）终端使用者隐瞒自己身份，进行不正确的输入。

（11）病毒侵入系统，破坏或修改了数据库软件。

面对数据库的安全威胁，需要通过有效措施来满足怎样的安全需求呢？这主要包括以下几个方面：

（1）防止非法数据访问。这是数据库安全最关键的需求之一。数据库管理系统必须根据用户或应用的授权来检查访问请求，以保证仅允许授权的用户访问数据库。

（2）防止推导。推导指的是用户通过授权访问的数据，经过推导得出的机密信息，而按照安全策略用户是无权访问该机密信息的。在统计数据库中需要防止用户从统计聚合信息中推导得到原始个体信息，特别是统计数据库容易受到推导问题的影响。

（3）保证数据库的完整性。该需求指的是保护数据库不受非授权的修改，以及不会因为病毒、系统中的错误等导致的存储数据破坏。

（4）保证数据的操作完整性。这个需求定位于在并发事务中保证数据库中数据的逻辑一致性。

(5) 数据的语义完整性。这个问题主要是在修改数据时保证新值在一定范围内以确保逻辑上的完整性。

(6) 审计和日志。为了保证数据库中数据的安全,一般要求数据库管理系统能够将所有的数据操作记录下来,以便事后调查和分析,追查入侵者或发现系统的安全弱点。

(7) 标识和认证。各种计算机系统的用户管理类似,使用的方法也非常类似。与其他系统一样,标识和认证也是数据库的第一道安全防线。

(8) 机密数据管理。数据库中的数据有部分可能是机密数据,也有可能全部是机密数据(如军队的数据库),而有些数据库中的数据全部是公开的数据。访问控制主要保证机密数据的保密性,仅允许授权用户的访问。

(9) 多级保护。多级保护表示一个安全需求的集合。在多级保护体系中,对不同数据项赋予不同的保密级别,然后根据数据项的密级给访问数据项的操作赋予不同的级别。

(10) 界限。界限的意义在于防止程序之间出现非授权的信息传递。

## 4.8.3 密码学在数据库安全上的应用

任何组织的成员数据库都是有价值的。一方面,需要把数据库分配给所有成员,使他们能够互相通信,交换想法。另一方面,如果把成员数据库分配给每个人,也会造成一些不希望的后果,例如,副本落入保险商人之手和其他恼人的垃圾邮件供应者之手等。

密码学可以改善这个问题。使用加密数据库,使得易于提取单个人的地址,而难于提取所有成员的邮件名单。

选用一个单向 Hash 算法和对称加密算法。数据库中的每个记录有两个字段:索引字段和数据字段。下面是一种方案:

索引字段是成员的姓,用单向 Hash 算法进行运算。数据字段是全名和成员的地址,用姓作为密钥对数据字段加密。除非知道这个人的姓,否则不可能解密数据字段。

这样加密之后,就容易搜索一个指定的姓。具体做法是:首先,对姓进行 Hash 运算,并在数据库的索引字段中搜寻散列值。如果匹配,那么这个人的姓就在数据库中。如果有几个匹配,那么就有几个人同姓。最后,对每个匹配的项,用姓作为密钥解密出全名和地址。

例如,有文献采用这种系统对 6000 个西班牙动词的字典进行保护。加密只引起很小的性能降低。附加的复杂性就是处理搜寻多个索引,但思想还是相同的。

这个系统的主要问题是当不知道怎么拼写他们的名字时,就不可能搜寻所要找的人。当然可以尝试各种拼法,直到找到正确的拼法为止。如果搜寻 Schneier 时,扫描所有以 Sch 开头的名字就不行。

但是,这种保护是不完善的。如果某个特别固执的保险商人通过试验所有可能的姓,就可能重构成员的数据库。如果他有电子数据库,他就可以把它作为一个可能的姓氏表来建立数据库。在计算机上做这件事的代价是要花费几个星期的时间,但是是可行的。在垃圾邮件社会中,做这种工作更难,而且“更难”很快会变成“太贵”。

## 4.9　本章重点和难点

本章重点是前 4 个小节。作为本科教学，建议课堂讲授第 4.1 节、第 4.2 节、第 4.3.1 节、第 4.3.3 节、第 4.4.1 节、第 4.4.2 节；研究生教学可适量增加第 4.4.5 节、第 4.4.6 节、第 4.4.9 节。建议后面几个小节的内容用于自学，以扩大阅读范围。

本章难点是清楚描述安全协议要解决的问题，以及解决这些问题采用的几种技术手段。

## 习题与思考题

1. 基本的安全协议有哪些？主要解决哪些安全问题？
2. 试述安全协议的形式化分析的重点。
3. 密钥交换协议的主要目的是什么？
4. 试述基本的安全协议所面临的主要攻击，以及为了防止这些攻击需要采取的措施。
5. 安全协议的第一步是认证。试述认证协议的演化进程。
6. 用 C 语言编写一个 SKEY 认证程序。
7. 试在信息认证中设计一种使第三方信任的协议形式。
8. 试举出一些秘密分割应用的实例。
9. 试设计一个带除名的秘密共享协议。
10. 试设计一种协议，对付字典式攻击。
11. 试分析随机数在对称密钥的管理和协议改进中的作用。
12. 试设计一个没有仲裁的密钥分割协议。
13. 举例说明密码学对于数据库安全性提高所起的作用。

# 第 5 章　抗攻击的安全协议

本章介绍安全协议的一些特殊应用要求，以及相应的技术手段。

本章共分为四个小节，第 5.1 节介绍抗攻击的密钥交换协议；第 5.3 节介绍抗攻击的认证协议；第 5.4 节是本章重点和难点分析。

## 5.1　对安全协议的设计和分析方法

### 5.1.1　对协议的典型攻击

对于安全协议的分析和设计，安全性是首要因素，而实际上可以认为是它抵抗某些攻击的能力。网络中的攻击主要分为两类，要求安全协议能够抵抗它们。

#### 1. 被动攻击

攻击者通过搭线窃听等方式，获得安全协议的部分消息（例如明文或者密文）。

攻击者可以通过收集明文-密文对来破译密码，获取一些秘密。例如：某次会话密钥，某个主体的私钥等。

被动攻击难以检测，因为它们并不会导致数据有任何改变。然而，防止这些攻击是可能的，因此，对付被动攻击的重点是采取加密传输等预防措施。

#### 2. 主动攻击

通常是指非法修改计算机网络中传输的报文。这些攻击涉及某些数据流的篡改或一个虚假数据流的产生。

主动攻击还可分为 4 类：伪装、重放、篡改和拒绝服务。例如：攻击者把截获的消息伪造、篡改或者有意地重放某些消息，以引起协议的不规则运行，从中获取一些秘密，甚至假冒某个合法的主体。数字签名保证了消息的不可篡改，抵御拒绝服务攻击必须采取其他措施。公钥密码体制下基于数字签名的协议遭受的首要威胁是假冒，其次是重放。要实施重放攻击，一般先要假冒身份。假冒是指攻击者冒充一个实体的身份；重放是指复制一个消息或报文，重新传送以产生未经授权的影响。

需要注意的是：攻击者不一定是协议外部的实体，他可能是一个合法用户，也可能是一个系统管理者。

通过对安全协议的深入剖析，大家发现单靠安全的密码算法是远远不够保证其安全的，安全协议本身的结构也对它的安全性产生显著的影响。有时攻击者可能不知道密钥，

但还是可以利用协议结构上的漏洞发起主动攻击。显然，这种隐患更值得警惕。

因此，对安全协议进行分析的目的，就是要使协议的安全性完全依赖于它所采用的密码体制的安全性，尽可能地消除由于协议本身的结构所造成的安全隐患，使得攻击者除了破译密钥之外，别无他法。

## 5.1.2 对协议安全性的分析

安全协议是许多分布式系统安全的基础，确保这些协议的安全运行是极为重要的。

大多数实用的安全协议只有为数不多的几个消息传递，其中每一个消息都是经过巧妙设计的，消息之间存在着复杂的相互作用和制约；同时，安全协议中使用了多种不同的密码体制。安全协议这种复杂的情况导致目前的许多安全协议存在安全缺陷。

造成协议存在安全缺陷的原因主要有两个：一是协议设计者误解或者采用了不恰当的技术；二是协议设计者对环境要求的安全需求研究不足。因此，对协议的安全性进行分析和研究是一个重要的课题。

目前，对安全协议进行分析的方法主要有两大类：一类是攻击检验方法；一类是形式化的分析方法。

所谓攻击检验方法就是搜集使用目前对协议有效的攻击方法，逐一对安全协议进行攻击，检验安全协议是否具有抵抗这些攻击的能力。协议攻击的目标通常有 3 种：第一是协议中采用的密码算法；第二是算法和协议中采用的密码技术；第三是协议本身。

在分析过程中主要使用自然语言和示意图，对安全协议所交换的消息进行剖析。这种分析方法往往是非常有效地，分析成功的关键在于攻击方法的选择。

## 5.1.3 安全协议的缺陷

尽管协议设计者尽可能在协议设计时回避可能出现的人为错误，但是安全协议在实际应用时仍会出现各种类型缺陷，产生的原因十分复杂，很难有一种通用的分类方法将安全协议的安全缺陷进行分类。

大致来说，安全协议的缺陷从来源上可分为两类：一类是由于设计不规范引发的；另一类是在具体执行时产生的。

但是这样分类太过笼统，S. GritzAlis 和 D. Spinellis 根据安全缺陷产生的原因和相应的攻击方法对安全缺陷进行了分类。

（1）**基本协议缺陷**：是由于在安全协议的设计中没有或很少防范攻击者而引发的协议缺陷。例如，对加密的消息签名，由于签名者并不一定知道被加密的消息内容，而且签名者的公钥是公开的，从而可使攻击者通过用他自己的签名替换原来的签名来伪装成发送者。

（2）**口令/密钥猜测缺陷**：这类缺陷产生的原因是用户往往从一些常用的词中选择其口令，从而导致攻击者能够进行口令猜测攻击；或者选取了不安全的伪随机数生成算法构造密钥，使攻击者能够恢复该密钥。口令猜测攻击可分为可检测的口令在线猜测攻击、不可检测的口令在线猜测攻击和可离线的口令猜测攻击 3 类。

（3）**陈旧消息缺陷**：主要是指协议设计中对消息的新鲜性没有充分考虑，从而使攻

击者能够进行消息重放攻击，包括消息源的攻击、消息目的攻击等。根据消息的来源与去向，陈旧消息攻击可分为消息来源攻击和消息目的地攻击。

(4) **并行会话缺陷**：协议对并行会话攻击缺乏防范，从而导致攻击者通过交换适当的协议消息能够获得所需要的信息。包括并行会话单角色缺陷、并行会话多角色缺陷等。

(5) **内部协议缺陷**：协议的可达性存在问题，协议的参与者中至少有一方不能完成所有必需的动作而导致的缺陷。

(6) **密码系统缺陷**：协议中使用的密码算法和密码协议导致协议不能完全满足所要求的机密性、认证等需求而产生的缺陷。

这种缺陷的分类囊括了安全协议缺陷来源的 3 个主要方面：

(1) 安全协议本身的缺陷。

(2) 协议需要依赖实施机制所产生的缺陷。

(3) 协议具体实施时产生的缺陷。

基本协议缺陷和并行会话缺陷属于协议本身的缺陷，即在假设协议所用到的密码算法及密码技术均是安全的前提下，协议仍旧存在的缺陷。口令/密钥猜测缺陷和陈旧消息缺陷属于协议需要依赖实施机制所产生的缺陷，因为这些缺陷的产生很大一部分都是依赖于实施采用的机制，也就是说，如果可以改善这些机制，那么就有可能避免这些缺陷。内部协议缺陷和密码系统缺陷属于具体实施时产生的缺陷，即在具体实施过程中会出错或者受到攻击。

## 5.1.4 安全协议的形式化分析

形式化的分析方法是采用各种形式化语言或者模型，为安全协议建立模型，并按照规定的假设和分析、验证方法证明协议的安全性。

在过去二十年中，形式化方法应用于广泛的领域：安全模型、流分析、安全协议分析、软件验证、硬件验证、体系结构分析、秘密信道分析等。

形式化方法在安全学界最大的成功是应用它分析安全协议。安全协议足够小，易于完成形式化分析，而且这些分析发现了很多以前不为所知的漏洞。

就其自身而言，形式化分析技术是有局限性的。协议系统的运行不是独立的，而是处于某种环境之下。系统的形式化说明是基于对系统环境的某种假设之上的。只有当假设成立时，证明才成立。所以，一旦某种假设不成立，一切证明就无从谈起。而入侵者只要违反了系统的假设，就可成功入侵系统。而且，即便明确给出假设的详细说明，也不可避免地会遗漏一些情况。

安全协议的安全性分析是一个很难解决的问题，许多看起来是安全的、并广泛应用的安全协议后来都发现存在安全缺陷。因此，研究人员想要得到从一开始就能证明协议安全性的工具。虽然这种工作很多都可以应用到一般的密码协议中去，但是研究的重点却毫无例外地放在认证和密钥交换上。

安全协议的形式化分析至少可以完成以下工作：

(1) 界定协议运行系统的边界。

(2) 更加准确描述系统的行为。

（3）更准确地定义系统的特性。

（4）证明系统在特定的假设前提下满足一定的特性。

#### 5.1.4.1　形式化分析方法的分类

形式化分析方法最早出现于 1989 年，Burrows、Abadi 和 Needham 提出的开创性的著作 A Logic of Authentication。此后出现的形式化分析方法大致可以分为 3 类：基于模型的计算方法，针对复杂系统的状态探查方法，符号操作观点。

基于模型的计算方法将攻击和被攻击的目标转化为一个游戏背景下的数学模型，通过对时间复杂度和概率模型的多项式时间归约，得到对某个数学难题的重大突破。通过证明这个重大突破的不可能发生来得到安全性的满足。

针对复杂系统的状态探查方法将协议模型转化成一个代数系统，表述参与者对协议知识的状态，通过对协议中主体在协议过程中的不同状态进行分析，并对其建立模型。然后分析不同状态之间的转移关系，通过转移关系之间的关联，找到不确定或不安全状态，分析某种状态的可达性，确认可能出现的问题。模型检验方法始于 20 世纪 90 年代，美国海军实验室的 NRL 协议分析器属于这种方法。1993 年由 McMillan KL 提出的符号模型检测法。1996 年，Lowe 首先采用 CSP（通信顺序进程）方法和模型校验技术对安全协议进行形式化分析。他应用 CSP 模型和 CSP 模型校验工具 FDR 分析了 NSPK 协议，并发现了一个 17 年来未知的攻击。

符号操作观点方法建立在理论计算机科学家对形式化方法的研究结果之上。在这种观点下，安全特性表述成一些可操作的抽象符号的集合，通过对主体的信念或知识的演变进行分析，获得主体最终的知识集合，并从知识集合中推导证明目标，如果推导成功，则协议能够满足证明目标中所规定的属性。

以 M. Burrows，M. Abadi 和 R. Needham 提出的 BAN 逻辑为先河，经过 20 多年的不懈努力，涌现出各种各样的安全协议形式化验证方法和相应的自动验证工具。

这些验证方法大致可以分为三类：信念逻辑（belief logics）方法、模型检测（model checking）方法和推理验证（inductive proofs）方法。

基于模型的计算方法和符号操作观点方法是通过证明安全协议能够满足它所声明的安全属性，从而证明此协议是安全的；第二种方法则是为了寻找安全协议的攻击剧本，从而证明此协议是不安全的。图 5-1 给出了安全协议的形式化分析方法的分类。

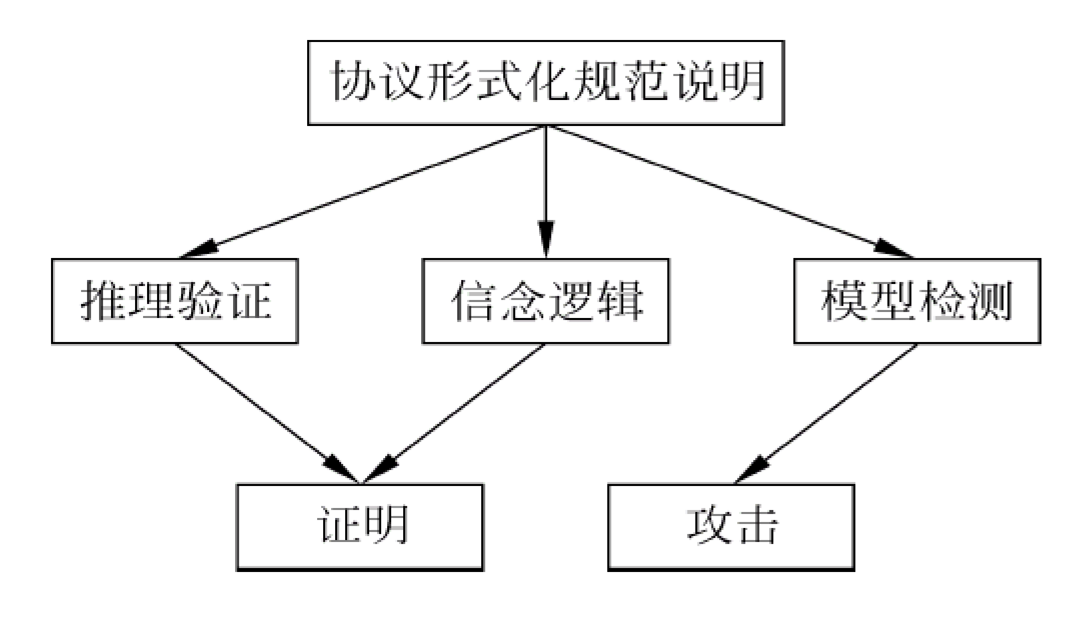



图 5-1　安全协议的形式化分析方法的分类

需要指出的是，由于安全协议本身的复杂性。目前并没有一种方法能够给出安全协议安全性的充分而且必要的理论证明。上述每一类方法都有不同的侧重点，或多或少地存在不足之处，在使用上述方法分析安全协议的时候。应当仔细分析协议的特点、应用环境和需求，综合使用这些分析方法，以得到比较合理的结果。

信念逻辑（belief logics）方法是使用逻辑系统，从用户接收和发送的消息出发，通过一系列推理公理来验证协议是否满足其安全说明。用于安全协议形式化分析的逻辑系统可以分为两类：一类是基于知识的，另一类是基于信念的。

模型检测（model checking）是目前使用非常频繁的安全协议形式化验证工具，并且使用这种方法取得了非常好的效果。模型刻划主体之间的相互作用，描述消息怎样发送和接收，特定的主体能够合成和传送哪些消息，以及在特定时间某个主体可能的行为等。简而言之，模型定义了一切可能发生的事件。事实证明，模型检查是分析安全协议的一种很成功的方法，基本方式是先建立运行协议的小系统模型（例如只有一个协议发起者和一个响应者），再加上能与协议交互的最具一般性的攻击者模型，然后使用状态搜索工具来检测系统是否可能进入不安全状态，即是否存在对协议的攻击。然而，作为验证安全协议正确性的方法，模型检查有严重局限性。由于模型检查器是通过穷举搜索模型中每个可能的状态，直到发现一个被入侵的特殊状态或者搜索完全部状态才会停止。因此，如果没有找到无限状态模型中这个特殊状态时，这个过程将永远不会停止。通常需要一个协议保证在有大量参与者参与，同时运行很多轮协议的时候，也是安全的，但无法使用有限状态模型完全获取该协议的安全性。

推理验证（inductive proofs）是用于证明安全协议满足某种属性的形式化方法。使用推理验证这种方法时，系统实现和系统规范都使用逻辑公式来描述，通过判断公式 I 是否蕴含公式 S 来验证系统实现与系统规范之间的一致性。验证的过程是逻辑公理系统中的推理证明过程。此方法的优点在于功能强大，适用面广，这是因为大部分问题都可以通过公理化转换为逻辑问题来解决。但是这种方法也有自身的缺陷，其缺陷在于使用这种方法并不能对所有问题都给出一个确定的答案，这是因为有些问题是不可判断的，所以使用推理验证时有可能永远不会停机，通常需要专家用户的交互，普通分析人员很难掌握这种分析方法。推理验证常用的工具有 HOL、PVS 和 Isabelle 等。

#### 5.1.4.2 入侵者模型和典型攻击类型

在协议运行的环境中，除了诚实的主体之外，还存在入侵者（intruder），为了形式化分析安全协议，对入侵者进行正确建模也是非常重要的。目前，主要有 3 种不同入侵者模型：Dolev-Yao 模型、spi 演算和概率模型。

Dolev-Yao 模型是 1983 年提出来的。目前，大多数形式化方法都是基于这种模型的。Dolev-Yao 模型包含以下假设前提：

（1）协议采用的密码算法是不可破解的。

（2）参加协议的主体除了有合法主体外，还有非法入侵者。

（3）与合法主体一样，入侵者具有相同知识和能力，如具有相同加密和解密密钥，可以得到网络中传递的任何消息，可以修改和发送他看到的任何消息。

Dolev-Yao 模型中的入侵者模型如图 5-2 所示。

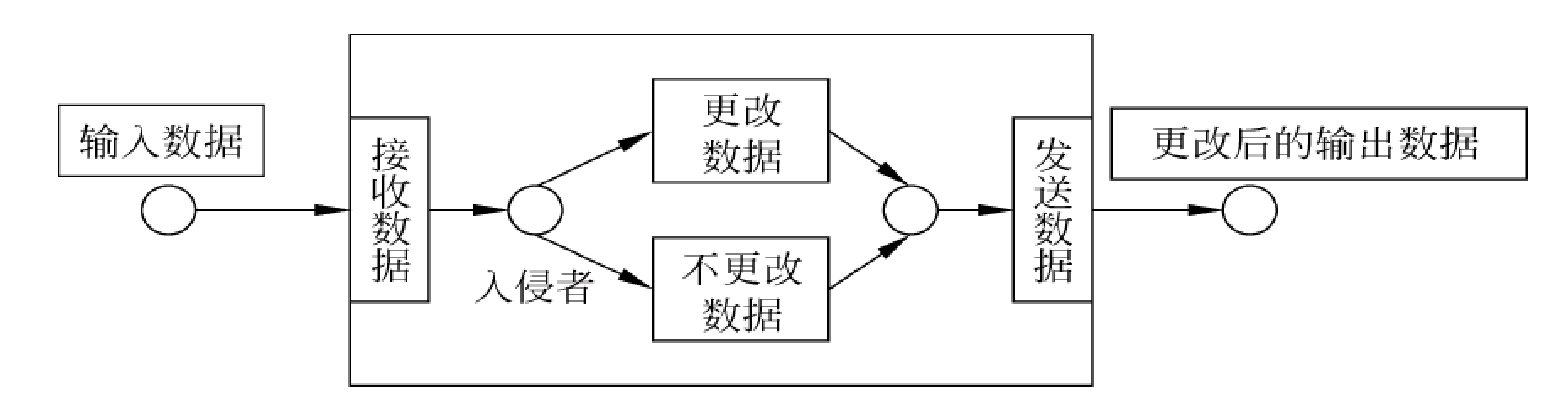



图 5-2　Dolev-Yao 模型中的入侵者模型

Dolev-Yao 模型是一个非常有用的抽象模型,虽然应用的范围非常广泛,但是它有自身的缺陷。Dolev-Yao 模型仅仅考虑了入侵者能够组合、重放消息以及使用已知密钥对解密消息,它不能够处理入侵者知道协议特定信息的情况,不能处理概率事件,例如入侵者试图猜测密钥。

spi 演算和概率模型请参阅有关文献[351]。

根据 Dolev-Yao 模型,安全协议在网络中可能遭受到的典型攻击包括:

(1) 消息重放攻击:在消息重放攻击中,攻击者预先记录某个协议先前运行中的某条消息,然后在该协议新的运行记录中重放该消息。

(2) 中间人攻击:攻击者将协议的某参与者向他提出的“挑战”交给另外的参与者回答,利用该参与者提供的“解密预言机”(random oracle)服务,产生可以通过挑战者认可的“应答”。

(3) 平行会话攻击:在攻击者的安排下,一个协议的两个或更多个运行并发执行,使得攻击者能够从一个运行中得到另外某个运行中的困难答案。如果用户 Bob 愿意几乎同时和 A 及 M 对话,那么平行会话攻击就常常是可行的。

(4) 反射攻击:一个诚实的主体给某个意定的通信方发送消息,用来让它完成密码操作时,M 截获该消息,修改底层通信协议处理的地址和身份信息,以便消息的产生者不会意识到反射回来的消息是“他自己”产生的。如果 M 成功了,那么消息产生者接受对问题的“回答”是自问自答,或者充当了攻击者的预言机。

(5) 拒绝服务攻击:在一个分布式的网络上,M 及其同伙可以向服务器提出多个连接或注册请求,这些请求由需要服务器维护的一些具有状态的 cookies 所组成。服务器的资源在维护同这些无用的 cookies 的连接的过程中耗尽,当合法的用户提出服务请求时,却会被拒绝。

(6) 交错攻击:在交错攻击中,某个协议的两次或多次运行在 M 的特意安排下按交织的方式执行。在这样一种攻击下,M 可以合成某条消息并发给某个运行中的某个主体,期望收到该主体的一个应答;而该应答可能对于另外某个运行中的另一个主体是有用的;在接下来的运行中,从前面运行中得到的应答可能会促使后面的主体对某个问题作出应答,而这个应答又恰好能用于第一个运行,如此交错进行。

#### 5.1.4.3　对密码协议形式化分析的基本途径

对密码协议形式化分析的基本途径有 4 种:

(1) 使用规范语言和验证工具建立协议模型和验证协议,它不是特别为密码协议分析设计的。

(2) 开发专家系统,协议设计者能够用它来调查研究不同的情况。

(3) 用分析知识和信任的逻辑,建立协议簇的需求模型。

(4) 开发形式化方法,它基于密码系统的代数重写项性质。

关于这4种途径的讨论,以及围绕它们的研究超出了本书的范围。这里略微介绍这个领域的主要方面。

第一种途径把密码学协议当作任何其他计算机程序,并试图证明它的正确性。然而证明正确性与证明安全性不同,并且这个方法对于发现许多缺陷的协议来说是行不通的。虽然这一种途径最早被广泛研究,但这个领域的大多数工作已经转向获得普及的第三种途径。

第二种途径是使用专家系统来确定协议是否能达到不合乎需要的状态(例如密钥的泄露)。虽然这种途径能够更好地识别缺陷,但它既不能保证安全性,又不能为开发攻击提供技术。它的好处在于决定协议是否包含已知的缺陷,但不可能发现未知的缺陷。

第三种途径是一种形式逻辑模型,是由 Michael Burrows、Martin Abadi 和 Roger Needham 首先发明,叫做 BAN 逻辑。BAN 逻辑是分析认证协议时用得最广泛的逻辑。它假设认证是完整性和新鲜度的函数,并使用逻辑规则来对贯穿协议的那些属性的双方进行跟踪。虽然已经提出了这种途径的许多变化和扩展,但大多数协议设计者仍在引用最初的研究。

BAN 逻辑并不提供安全性证明,它只能推出认证。它具有容易使用的简单、明了的逻辑,对于发现缺陷仍然有用。BAN 逻辑中的一些命题有:

(1) A 相信 X(A 装作好像 X 是正确的)。

(2) A 看 X(某些人已经把包含 X 的消息发给 A,A 可能在解密消息后才能够读和重复 X)。

(3) A 说 X(在某一时间,A 发送包括命题 X 的消息。不知道的是,消息在多早以前曾被发送过,或是在协议当前运行期间发送的。已经知道,当 A 说 X 时,A 相信 X)。

(4) X 是新的(在当前运行协议以前,X 在任何时间没有把消息发送出去)等。

BAN 逻辑也为协议中有关信任理由提供规则。这些规则能够用到协议的逻辑命题,用来证明事情或回答有关协议的问题。例如,消息内涵的规则是:

(1) 如果 A 相信 A 和 B 共享密钥 K,A 看见用 K 加密的 X,而 A 没有用 K 加密 X,那么 A 相信 B 曾经说过 X。

另一个规则是只以当时为限的验证规则:

(2) 如果 A 相信 X 只在最近被发送,并且 B 曾经说过 X,那么 A 就认为 B 相信 X。

用 BAN 逻辑进行分析分4步:

(1) 采用以前描述的命题,把协议转换为理想化形式。

(2) 加上有关协议初始状态的所有假设。

(3) 把逻辑公式放到命题中:在每个命题后断言系统的状态。

(4) 为了发现协议各方持有的信任,运用逻辑基本原理去断言和假设。

BAN 逻辑的作者"把理想化的协议看作比在文献中发现传统的描述更清楚和更完善的规范…"。其他协议没有这种印记,并因为它不可能正确反映实际协议而批评这个步骤。其他批评试图表明,BAN 逻辑可能推导出关于协议明显错误的特征,并且 BAN 逻辑只涉及信任,而与安全性无关。

尽管有这些批评,BAN 逻辑仍是成功的。它已经在几种协议中发现缺陷,这些协议包括 Needham-Schroeder 和一个早期 CCITTX.509 协议草案。它已经发现很多协议中的冗余,这些协议包括 Yahalom、Need-ham-Schroeder 和 Kerberos。许多人的文章使用 BAN 逻辑,声称他们协议的安全性。

其他逻辑系统也有公布,一些设计成为 BAN 逻辑的扩展,另一些是基于 BAN 逻辑去改进发现的弱点。

密码协议分析的第四种途径是把协议当作一个代数系统模型,表示有关协议参与者了解的状态,然后分析某种状态的可达性。这种途径没有像形式逻辑那样引起更多的注意,但情况正在改变。它首先由 Michael Merritt 使用,他证明代数模型可用来分析密码协议。

美国海军研究试验室(Navy Research Lab,NRL)的协议分析器可能是这些技术中最成功的应用,用来在各种协议中寻找新的和已知的缺陷。这台协议分析器定义了下面的行为:

(1) 接收(B,A,M,N)(B 在 N 地附近,接收消息 M 作为来自 A 的消息)。

(2) 获悉(E,M)(E 获悉 M)。

(3) 发送(A,B,Q,M)(根据查询 Q,A 发送 M 给 B)。

(4) 请求(B,A,Q,N)(B 在 N 地附近,发送 Q 给 A)。

从这些行为中,可以确定需求,例如:

(1) 如果 B 在过去某些点接收到从 A 来的消息 M,那么 E 在过去某些点没有获悉 M。

(2) 如果 B 在他的 N 地附近接收到从 A 来的消息 M,A 给 B 发送 M 作为 B 在 N 地附近查询的响应。

为了使用 NRL 协议分析器,必须按以前的结构规定协议。分析有 4 个步骤:为诚实的参与者定义传送;描述对所有诚实和不诚实参与者可得到的操作;描述基本协议构造部件;描述还原规则。

这里表述的所有要点是已知的协议要与它的需求相符,采用像 NRL 协议分析器这样的工具,最终会产生一个能够证明是安全的协议。

### 5.1.5 安全协议的设计原则

安全协议的设计极易出错,即使只讨论安全协议中最基本的认证协议,其中参加协议的主体只有两三个,交换的消息只有 3～5 条,设计一个正确的、符合认证目标的、没有冗余的认证协议也十分困难。因此,多年以来,为了应对这一挑战,人们设计了不同种类的形式化分析方法,投入了大量的精力,取得了可喜的成果。

安全协议设计与分析的困难性在于：

(1) 安全目标本身的微妙性。例如，表面上十分简单的“认证目标”，实际上十分微妙。

(2) 协议运行环境的复杂性。实际上，当安全协议运行在一个十分复杂的公开环境时，攻击者处处存在。我们必须形式化地刻划安全协议的运行环境，这是一项艰巨的任务。

(3) 攻击者模型的复杂性。我们必须形式化地描述攻击者的能力，对攻击者和攻击行为进行分类和形式化分析。

(4) 安全协议本身具有“高并发性”特点。因此，安全协议的分析变得更加复杂并具有挑战性。

针对不同的安全协议，许多研究者提出了一些具体的安全协议设计原则。例如，Anderson 和 Needham 提出的设计公钥安全协议原则，能够有效指导公钥安全协议的设计。

**原则 1** 签名先于加密。如果不这样做，而是对于一个加密数据进行签名，那么不能认为签名者知道数据明文。第三方也就不能认为这个签名是有效的。

**原则 2** 注意各个实体是如何区分的。如果可能，尽量避免对不同目的(例如签名和加密)使用相同密钥；另外要确定能够区分同一个协议的不同协议轮。

**原则 3** 当签名和加密数据的时候，要注意自己不要被对方当作先知(Oracle)，使得能够发起 Oracle 攻击。

**原则 4** 说明协议中所有数据。例如，哪些数据是防止协议模棱两可的，哪些数据是提供冗余的，哪些数据是使得计算机具有复杂性的等。确信协议需要的冗余性的基础在应用中是鲁棒的，确信任何附加的数据不能以任何方式来攻击协议。

**原则 5** 不要假设任何其他人的秘密(除了一个 CA 的秘密)是保密的。

**原则 6** 不要假设收到的消息一定来自于某个特定的主体，除非能以某种方法检测这一点。

又如，Martin Abadi 和 Roger Needham 提出设计协议的 11 条原则，能够指导安全协议的设计，尽量减少安全协议设计的漏洞。虽然，这些原则对于安全协议设计的安全性不是完备和充分的，但是它们是非常有用的，能够指导协议的设计者简化协议的设计，并且能够防范已知的一些协议漏洞。

Martin. Abadi 和 Roger Needham 对安全协议设计的原则进行了总结，遵守这些原则有助于简化协议和避免一些早已公开的协议错误。

两个基本设计原则如下：

(1) 一个消息所要传达的内容的所有要素必须明确地表达在该消息中，接收者无需从上下文推得某些信息即可恢复该消息的全部意义，否则该消息可能会被替代用来进行欺骗。

(2) 一个消息可被处理的条件必须明确地列出，以便其他回顾该协议设计的人判断这些条件能否接收。

对上述原则进行分析和扩展，可以得到以下几个具体设计要点：

(1) 若主体身份对消息的意义是不可缺少的,则在该消息中明确包含主体的身份可以防止某些攻击。

(2) 使用一次性随机数(nonce)或挑战(challellge),而不是时间戳(time stamp)。

在协议设计中,时间戳和 nonce 被用于保证消息的新鲜性,以防止消息重放。但两者之间有很重要的区别:使用时间戳时一般要求各主体的时钟同步,但时间戳并不和某个主体直接关联,任何一个主体产生的时间戳都能被其他主体用来检验消息的新鲜性。nonce 则是某个主体产生的随机数,一个主体只能根据它自己所产生的 nonce 来检验消息的新鲜性。此外,时间戳不具有唯一性,它通常有一个有效范围,只要它位于这个有效范围内,主体都接受它的新鲜性。nonce 则具有唯一性,任何一个主体在两次会话中产生的临时值在很长一段时间内不可能相同。

挑战通常是不可预测的且未使用过的随机数,而响应则是其函数,采用挑战的目的为防止重放攻击;然而,随机数只能保证及时性,不能防止利用两个并行协议构造的攻击。因此,挑战必须包含完整的信息,如预期中挑战的响应者名字,挑战的发起者发起该挑战的目的,这些可用加密捆绑在一起。

(3) 最少的初始化假设。在进行协议设计前,通常需要对网络环境作出分析,提出合适的初始化假设。

例如,CA 是可信赖或信道是可信赖的,或者各实体都拥有自己的证书,等等。但是,初始化假设并非越多越好,因为有些假设可能本身存在不确定因素,或者根本经不起严格推理,所以应尽可能减少初始化假设数目。

(4) 消息应具有明确的意义,且必须明确对消息进行加密的目的。

协议中每一条消息都有其意义和作用,对消息的解释应该依赖于消息本身。即使有一种合适的形式化语言可以描述某条消息,也应该可以用一句话来描述其内容,即可以将消息由符号转换为更易理解的语句。

在安全协议设计中,加密可以用来保证消息的机密性、指示消息的来源、捆绑消息的各个部分,以及产生随机数等。因此,明确加密所起的作用对于理解具体的协议很重要。

在安全协议的分析与设计中,以上这些原则能够帮助协议分析者发现协议的漏洞,从而提出改进方法;协议设计者可以遵循这些原则来设计协议,避免一些已发现的协议错误。

## 5.2　抗攻击的密钥交换协议

### 5.2.1　中间人攻击

在密钥交换协议中,A 和 B 由于需要进行秘密信息的交互,需要交换密钥,此时,窃听者 E 除了试图破译公开密钥算法或者尝试对密文作唯密文攻击之外,没有更好的办法。但是,对于恶意的主动攻击者 M 来说,比 E 更有能力,他不仅能监听 A 和 B 之间的信息、还能修改信息、删除信息、并能产生全新的信息。当 M 同 A 谈话时,他能模仿 B,他

也能模仿 A 同 B 谈话。

下面介绍这种中间人攻击是怎样生效的：

(1) A 将她的公开密钥传送给 B。M 截取这个密钥，并将自己的公开密钥传送给 B。

(2) B 将他的公开密钥传送给 A。M 截取这个密钥，并将自己的公开密钥传送给 A。

(3) 当 A 将用"B"的公开密钥加密的信息传送给 B 时，M 截取它。由于信息实际上是用 M 的公开密钥加密的，他就用自己的私钥解密，再用 B 的公开密钥对信息重新加密，并将它传送给 B。

(4) 当 B 将用"A"的公开密钥加密的信息传送给 A 时，M 截取它。由于信息实际上是用 M 的公开密钥加密的，他就用自己的私钥解密，再用 A 的公开密钥对信息重新加密，并将它传送给 A。

即使 A 和 B 的公开密钥存储在数据库中，这种攻击也是可行的。M 可以截取 A 的数据库查询，并用自己的公开密钥代替 B 的公开密钥，对 B 也可以做同样的事情，用自己的公开密钥代替 A 的公开密钥。他也能秘密地侵入数据库，用自己的密钥代替 A 和 B 的密钥。接下来就简单地等着 A 和 B 互相谈话，然后截取和修改信息，他就成功了！

"中间人攻击"是可行的，因为 A 和 B 无法验证他们正在作的互相交谈。只要 M 没有导致任何值得注意的网络延迟，他们两人就没有办法知道有人正在他们中间阅读他们自认为是秘密的消息。

## 5.2.2 阻止中间人攻击的联锁协议

由 Ron Rivest 和 Adi Shamir 发明的联锁协议是阻止"中间人攻击"的一种办法。下面是这个协议的工作过程：

(1) A 将她的公开密钥传送给 B。

(2) B 将他的公开密钥传送给 A。

(3) A 用 B 的公开密钥加密她的报文，并将加密报文的一半传送给 B。

(4) B 用 A 的公开密钥加密他的报文，并将加密报文的一半传送给 A。

(5) A 将加密的另一半报文传送给 B。

(6) B 将 A 的两半报文合在一起，并用他的私钥解密；B 将他加密的另一半报文传送给 A。

(7) A 将 B 两半报文合在一起，并用她的私钥解密。

该协议过程如图 5-3 所示。

这里要注意的是：只有报文的一半，没有另一半，报文是毫无用处的。B 只有到步骤(6)才能读 A 的报文，A 只有到步骤(7)才能读 B 的报文。有很多办法实现它：

(1) 如果采用分组加密算法，每一分组的一半(例如每隔一比特)能在每半个报文中发送。

(2) 报文的解密依赖于初始矢量，初始矢量可以在报文的另一半中发送。

(3) 首先发送的一半报文可能是加密报文的单向 Hash 算法(参看 3.2.3 节)，并且加密报文本身可能是另一半。

为了了解这样做是怎样对 M 制造麻烦的，让我们再看看他破坏协议的过程。M 能够在

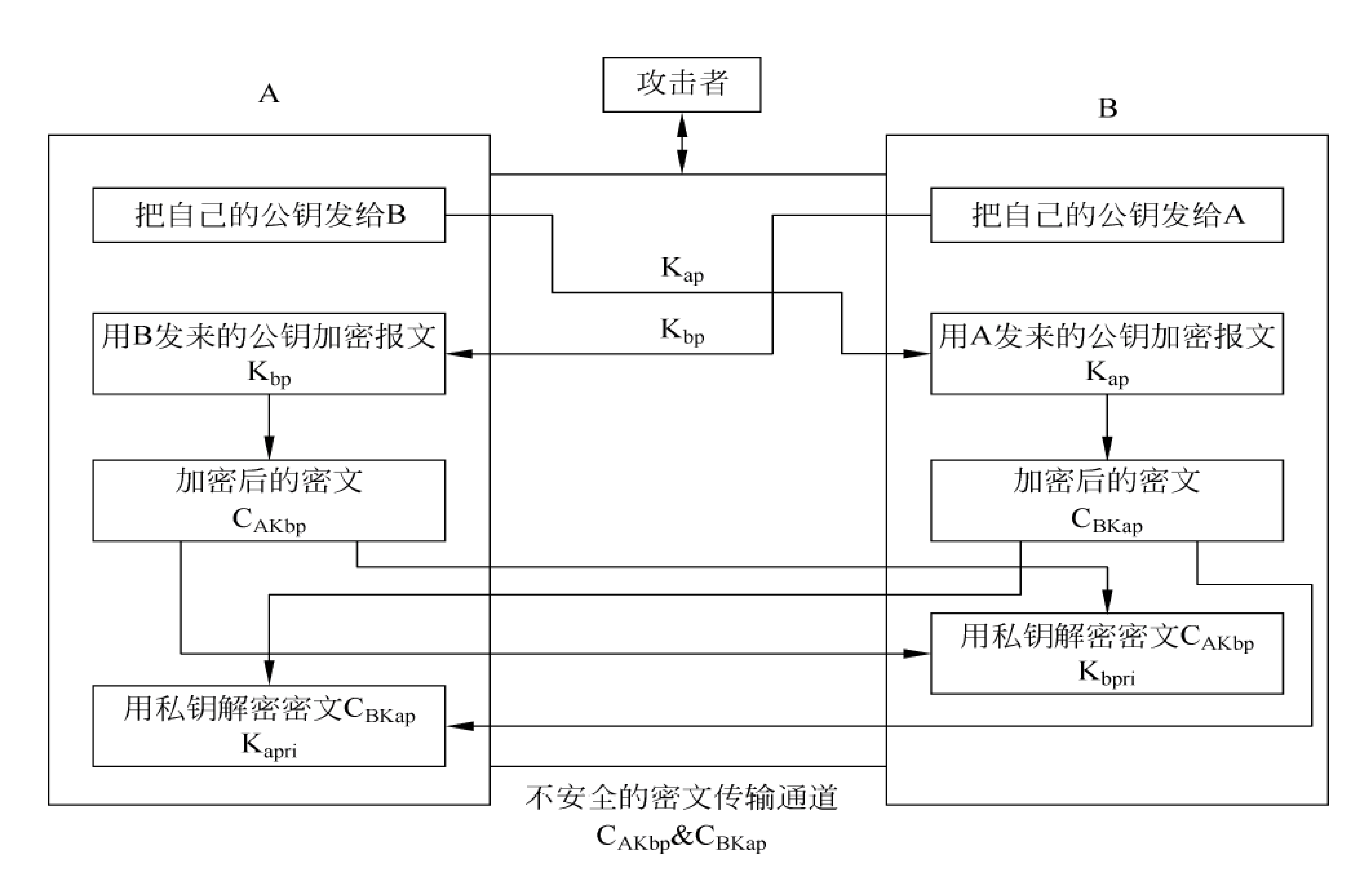



图 5-3　联锁协议

步骤(1)和步骤(2)中用自己的公开密钥代替 A 和 B 公开密钥。但现在，当他在步骤(3)截取 A 的一半报文时，他不能用自己的私钥对报文解密，然后用 B 的公开密钥再加密，他不得不虚构一个完全不同的新报文，并将它的一半发送给 B。当他在步骤(4)截取 B 给 A 的一半报文时，他有同样的问题，他不能用自己的私钥解密，并用 A 的公开密钥再加密，他又不得不虚构一个完全不同的新报文，并将它的一半发送给 A。当他在步骤(5)和步骤(6)截取到实际报文的另一半时，他再去把自己虚构的新报文改回来，就太迟了。A 和 B 之间的会话必定是完全不同的。

M 也可以不用这种办法。如果他非常了解 A 和 B，他就可以模仿他们之中的一个人同另一人通话，他们绝不会想到正受到欺骗。但这样做肯定比坐在他们之间截取和读他们的报文更难。

### 5.2.3　使用数字签名的密钥交换协议

在会话密钥交换协议期间，采用数字签名也能防止“中间人攻击”。比如，T 对 A 和 B 的公开密钥签名，签名的密钥包括一个已签名的所有权证书。当 A 和 B 收到密钥时，每人都能验证 T 的签名。这样，他们就知道公开密钥是哪个人的。密钥的交换就能完成了。协议过程如图 5-4 所示。

M 会遇到严重的阻力。他不能假冒 B 或者 A，因为他不知道他们的私钥。他也不能用自己的公开密钥代替 A 和 B 两人的公开密钥，因为他由 T 签名的证书，是为 M 自己签发的。他所能做的事情就是窃听往来的加密报文，或者破坏通信线路，阻止 A 和 B 谈话。

这个协议也动用 T，但 KDC 遭受损害的风险比较小。假设 M 危及到 T 的安全(例如侵入 KDC)，他所得到的只是 T 的私钥。这个密钥使他仅能对新的密钥签名；不会使 M

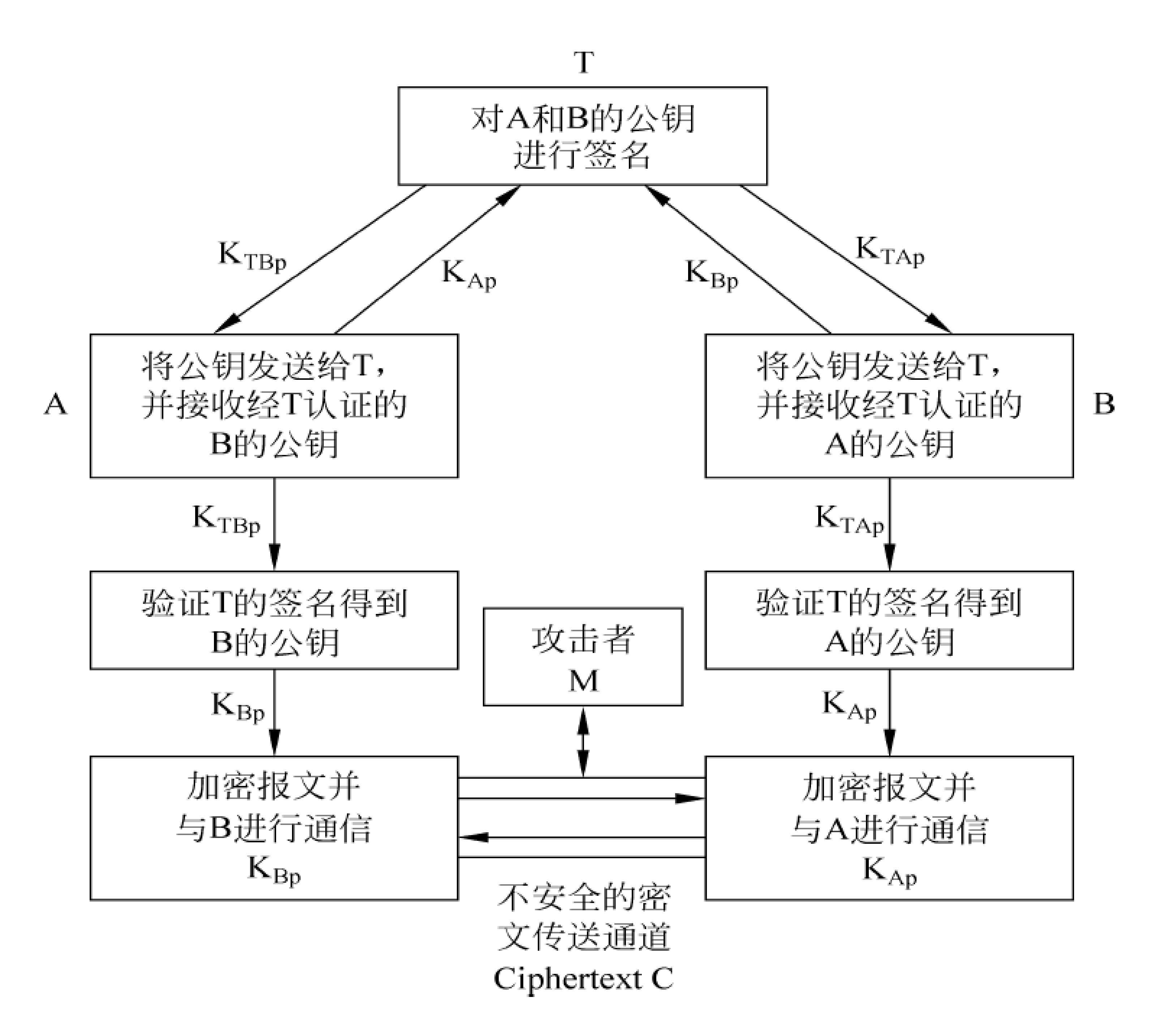



图 5-4　数字签名的密钥交换

能够对任何会话密钥解密，或者读取任何报文。为了能够读往来的报文，M 不得不冒充网络上的某个用户，并且欺骗合法用户用他的假的公开密钥加密报文。

M 可以采用下述方法发起这种攻击：持有 T 的私钥，他能够产生假的签名密钥去欺骗 A 和 B。然后 M 就可以在数据库中交换他们真正的签名密钥，或者截取用户向数据库的请求，并用他的假密钥代替。这使他能够实施“中间人攻击”，并读取他人的通信。

这种攻击是可行的，但条件是 M 必须能够截取和修改信息。在一些网络中，截取和修改报文比被动地坐在网络旁读取往来的报文更难。在广播信道上，如无线网中，几乎不可能用其他报文来替代某个报文(否则整个网络可能被堵塞)。在网络中做这种事要容易些，并且随着时间的推移变得越来越容易，例如 IP 欺骗、路由攻击等。

## 5.2.4　密钥和报文传输协议

在下面的协议中，A 在没有任何密钥交换协议的情况下，将报文 M 传送给 B：

(1) A 产生一个随机会话密钥 K，并用 K 加密 M：$E_K(M)$。

(2) A 从数据库中得到 B 的公开密钥。

(3) A 用 B 的公开密钥加密 K：$E_B(K)$。

(4) A 将加密的报文和加密的会话密钥传送给 B：$E_K(M),E_B(K)\rightarrow B$。为了增加安全性，防止“中间人攻击”，A 可对传输签名。

(5) B 用自己的私钥将 A 的会话密钥 K 解密。

(6) B 用会话密钥 K 将 A 的报文解密。

该协议过程如图 5-5 所示。

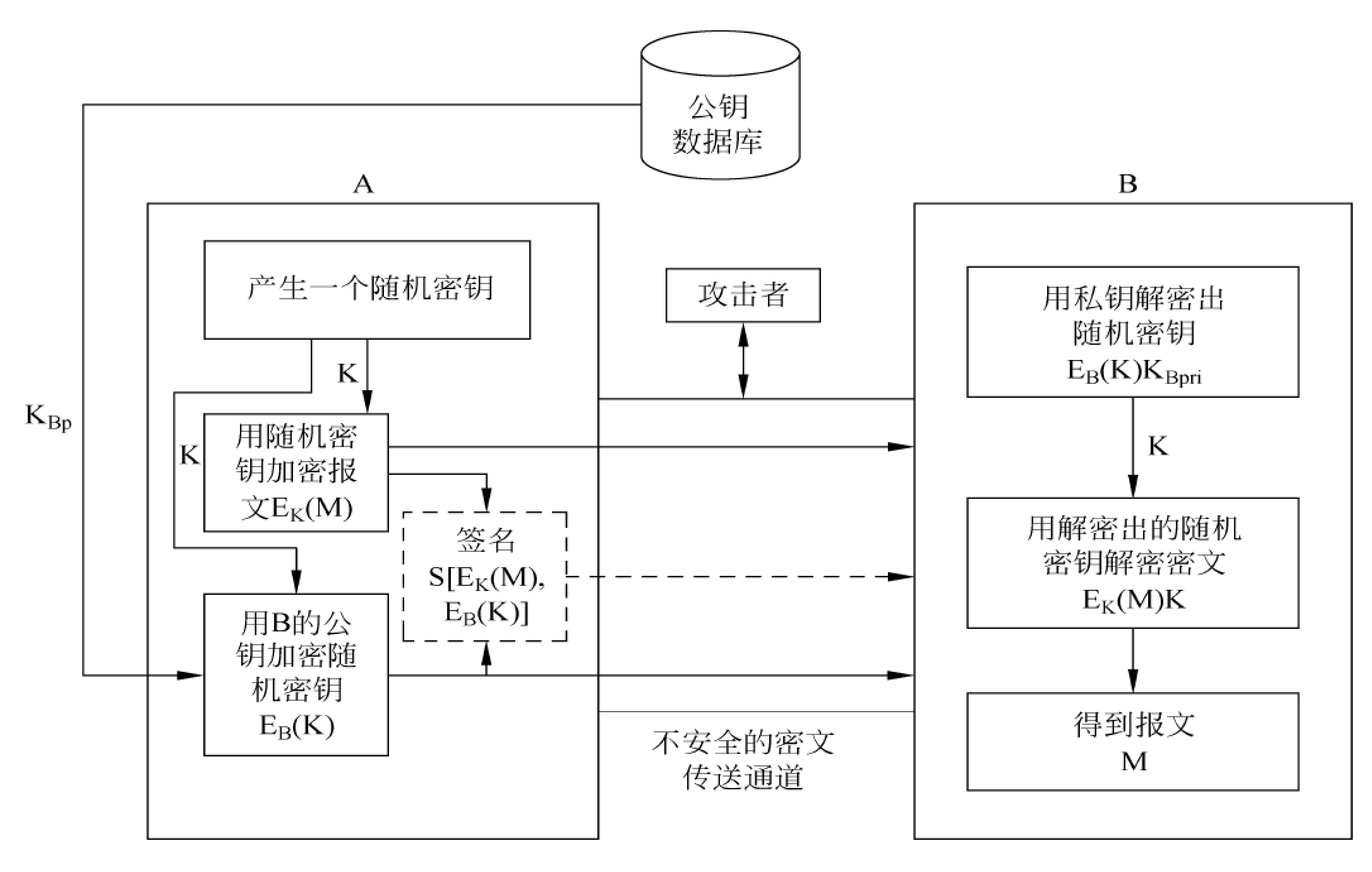



图 5-5　密钥和报文传输

这个混合系统表示，公开密钥密码是怎样经常用于通信系统的。它可以和数字签名、时间标记以及任何其他安全协议组合在一起使用。

### 5.2.5　网络存储应用中的密钥和报文广播协议

对于 A 把加密的报文传送给另外几个人的情况，是在下面这个例子中，A 把加密报文传送给 B、C 和 D。

（1）A 产生一个随机会话密钥 K，并用 K 加密报文 M：$E_K(M)$。

（2）A 从数据库中得到 B、C 和 D 的公开密钥：$E_B$、$E_C$、$E_D$。

（3）A 用 B 的公开密钥 $E_B$ 加密 K，用 C 的公开密钥 $E_C$ 加密 K，用 D 的公开密钥 $E_D$ 加密 K：$E_B(K)$、$E_C(K)$、$E_D(K)$。

（4）A 广播加密的报文和所有加密的密钥，并传送给要接收的人：$E_B(K)$、$E_C(K)$、$E_D(K)$、$E_K(M)$。

这个例子中，只有 B、C 和 D 可以分别用他们的私钥解密 K；只有 B、C 和 D 可以分别用解密出的 K 再解密 A 的报文。

上述过程如图 5-6 所示。

这个协议可以在存储转发网络上实现。中央服务器将 A 的报文，连同加密后的密钥一起转发给 B、C 和 D。服务器不一定需要是安全的或者可信的，因为它不可能对任何报文解密。

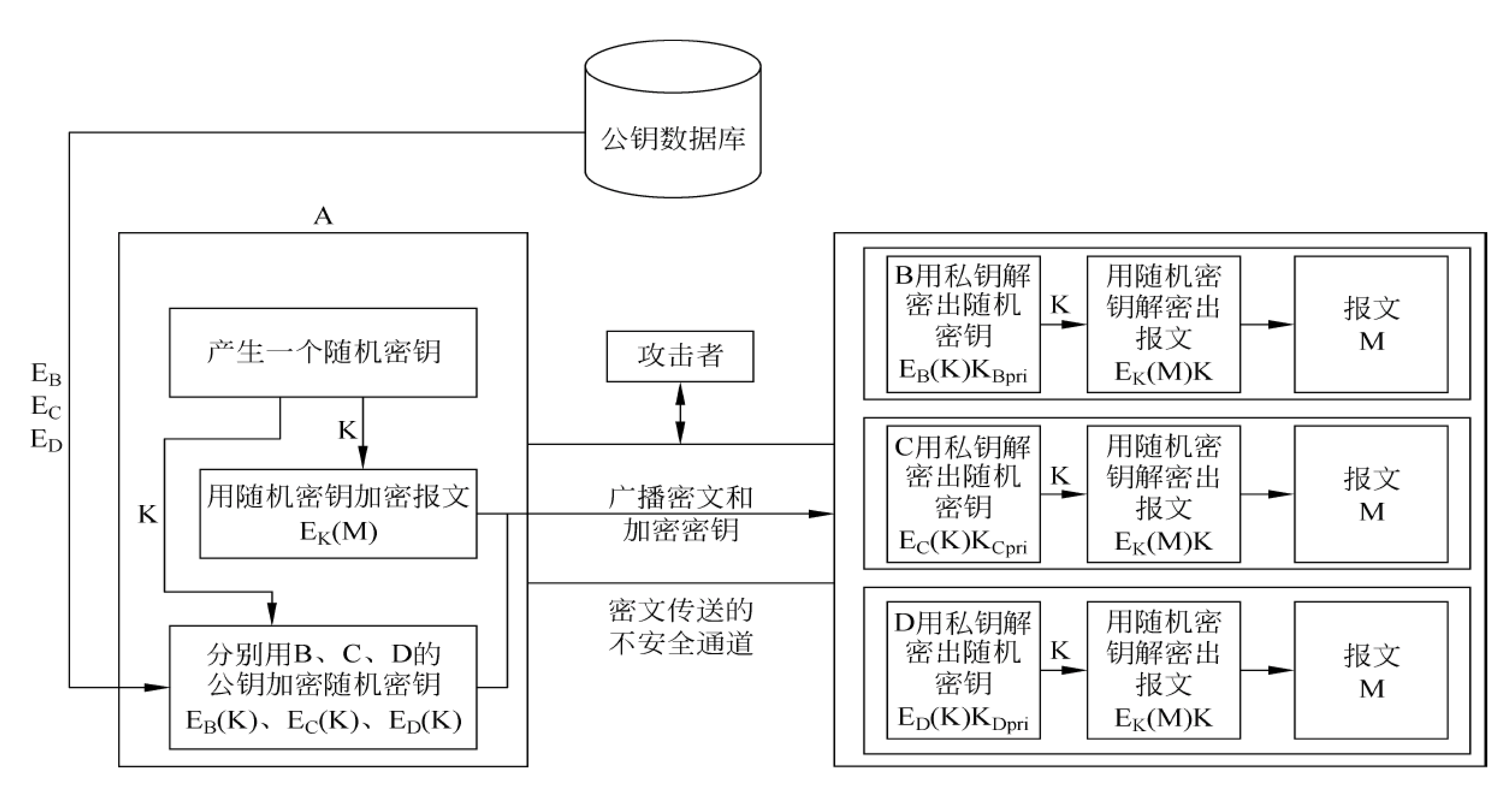



图 5-6　密钥和报文广播

# 5.3　抗攻击的认证协议

## 5.3.1　对于认证协议的攻击举例

在讨论抗攻击的认证协议之前，首先了解一下，目前有哪些攻击类型是针对认证协议而发起的。

### 5.3.1.1　采用对称密码和 T 的认证协议

这个协议由 Roger Needham 和 Michael Schroeder 发明，也称为 Needham-Schroeder 协议。

(1) A 将由自己的名字，B 的名字和随机数 $R_A$ 组成的报文传给 T。

$$(A, B, R_A) \rightarrow T$$

(2) T 产生一个随机会话密钥 K。他用与 B 共享的秘密密钥对随机会话密钥 K 和 A 名字组成的报文加密。然后用他和 A 共享的秘密密钥对 A 的随机值、B 的名字、会话密钥 K 和已加密的报文进行加密，最后，将加密的报文传送给 A。

$$E_A(R_A, B, K, E_B(K, A)) \rightarrow A$$

(3) A 将报文解密并提取 K。她确认 $R_A$ 与她在步骤(1)中发送给 T 的一样。然后她将 T 用 B 的密钥加密的报文发送给 B。

$$E_B(K, A) \rightarrow B$$

(4) B 对报文解密并提取 K，然后产生另一个随机数 $R_B$。他用 K 加密它并将它发送给 A。

$$E_K(R_B) \rightarrow A$$

(5) A 用 K 将报文解密,产生 $R_B-1$ 并用 K 对它加密,然后将报文发回给 B。

$$E_K(R_B-1) \rightarrow B$$

(6) B 用 K 对信息解密,并验证它是 $R_B-1$。

上述这些围绕 $R_A$、$R_B$、$R_B-1$ 的信息用来防止重放攻击。

在重放攻击中,M 记录旧的报文,在以后再使用它们以达到破坏协议的目的。本协议中在步骤(2)中 $R_A$ 的出现使 A 确认 T 的报文是合法的,并且不是以前协议的重放。在步骤(5),当 A 成功地解密 $R_B$,并将 $R_B-1$ 送回给 B 之后,B 确认 A 的报文不是早期协议执行的重放。

这个协议的主要安全漏洞是旧的会话密钥仍有价值。如果 M 可以存取旧的密钥 K,他就可以成功地实施一次攻击。他所做的工作是,记录 A 在步骤(3)发送给 B 的报文。然后,一旦他有 K,他就可以假装是 A:

(1) M 发送给 B 下面的信息。

$$E_B(K, A) \rightarrow B$$

(2) B 提取 K,产生 $R_B$,并发送给 A。

$$E_K(R_B) \rightarrow A$$

(3) M 截取此报文,用 K 对它解密,并发送给 B。

$$E_K(R_B-1) \rightarrow B$$

(4) B 验证 A 的报文是 $R_B-1$。

到此为止,M 成功地使 B 确认他就是 A 了。

使用时间标记的一种更强的协议能够击败这种攻击。在步骤(2)中,一个时间标记被附到用 B 的密钥加密的 T 的信息中:$E_B(K, A, T)$。时间标记需要一个安全的和精确的系统时钟,这对系统本身来说是另一个难题。

如果 T 与 A 共享的密钥 $K_A$ 泄露了,后果是非常严重的。M 能够用它获得同 B 交谈的会话密钥(或他想要交谈的其他任何人的会话密钥)。甚至,在 A 更换她的密钥后 M 还能够继续做这种事情。

#### 5.3.1.2 Otway-Rees 协议及其"类型缺陷"型攻击

Otway-Rees 协议使用对称密码。主要过程如下:

(1) A 产生一个报文,此报文包括一个索引号 I、她的名字 A、B 的名字和一个随机数 $R_A$,用她和 T 共享的密钥 $E_A$ 对此报文加密,她将索引号、她的名字和 B 的名字与她加密的报文一起发送给 B。

$$I, A, B, E_A(R_A, I, A, B) \rightarrow B$$

(2) B 产生一个报文,此报文包括一新的随机数 $R_B$、索引号 I、A 的名字和 B 的名字。用他与 T 共享的密钥 $E_B$ 对此报文加密。他将 A 的加密报文、索引号、A 的名字、B 的名字与他加密的报文一起发送给 T。

$$I, A, B, E_A(R_A, I, A, B), E_B(R_B, I, A, B) \rightarrow T$$

(3) T 产生一个随机会话密钥 K,然后,产生两个报文。一个是用他与 A 共享的密钥

对 A 的随机数和会话密钥加密，另一个是用与 B 共享的密钥对 B 的随机数和会话密钥加密。他将这两个报文与索引号一起发送给 B。

$$I, E_A(R_A, K), E_B(R_B, K) \rightarrow B$$

(4) B 将用 A 的密钥加密的报文连同索引号一起发送给 A。

$$I, E_A(R_A, K) \rightarrow A$$

(5) A 解密报文，恢复出她的密钥和随机数，然后她确认协议中的索引号和随机数都没有改变。

假设所有随机数都匹配，并且按照这种方法索引号没有改变，A 和 B 现在相互确认对方的身份，他们就建立了一个密钥可以用于通信了。

"类型缺陷(type flaw)"型攻击的特点是利用认证协议实现时的固定格式对协议进行攻击。假定在上述认证协议中，M 的长度是 64 比特，A 和 B 的长度各为 32 比特，$K_{ab}$ 的长度为 128 比特。用户 A 在发起协议之后，预期在协议的步骤(4)可以收回他在协议步骤(1)建立的临时值 $N_a$，以及认证服务器 S 为他分配的会话密钥 $K_{ab}$。这时，攻击者 P 可以冒充 B，重放消息(1)中的加密分量，将它作为消息(4)中的加密分量发送给 A。攻击过程如下：

(1) $A \rightarrow P(B): M, A, B, \{N_a, M, A, B\}Kas$

(4) $P(B) \rightarrow A: M, \{N_a, M, A, B\}Kas$

收到消息(4)之后，A 解密 $\{N_a, M, A, B\}Kas$，校验 $N_a$ 的正确性，如果无误则根据协议接收(M,A,B)为新的会话密钥。但是，(M,A,B)是攻击者 P 可以掌握的公开消息。因此，攻击者成功地进行了攻击，今后可以通过会话密钥(M,A,B)监听 A 和 B 之间的会话。

类似地，攻击者可以冒充认证服务器 S 攻击 Otway-Rees 协议。这时，P 只需将消息(2)中的加密分量重放给 B 即可。攻击过程如下：

(1) $A \rightarrow B: M, A, B, \{N_a, M, A, B\}Kas$

(2) $B \rightarrow P(S): M, A, B, \{N_a, M, A, B\}Kas, \{N_b, M, A, B\}Kbs$

(3) $P(S) \rightarrow B: M, \{N_a, M, A, B\}Kas, \{N_b, M, A, B\}Kbs$

(4) $B \rightarrow A: M, \{N_a, M, A, B\}Kas$

这样，攻击者就成功地进行了攻击，今后就可以通过会话密钥(M,A,B)监听 A 和 B 之间的会话。

## 5.3.2 时间戳服务

时间戳(time-stamp)是一个经加密后形成的凭证文档，它包括 3 个部分：需加时间戳的文件摘要(digest)、DTS 收到文件的日期和时间以及 DTS 的数字签名。其产生过程一般为：用户首先将需要加时间戳的文件用 Hash 值进行编码加密形成摘要，然后将该摘要发送到 DTS，DTS 在加入了收到文件摘要的日期和时间信息后再对该文件加密(数字签名)，然后送回用户。

时间戳服务是网上电子商务安全服务项目之一，它能提供对电子文件的日期和时间信息的安全保护。其实现原理如图 5-7 所示。

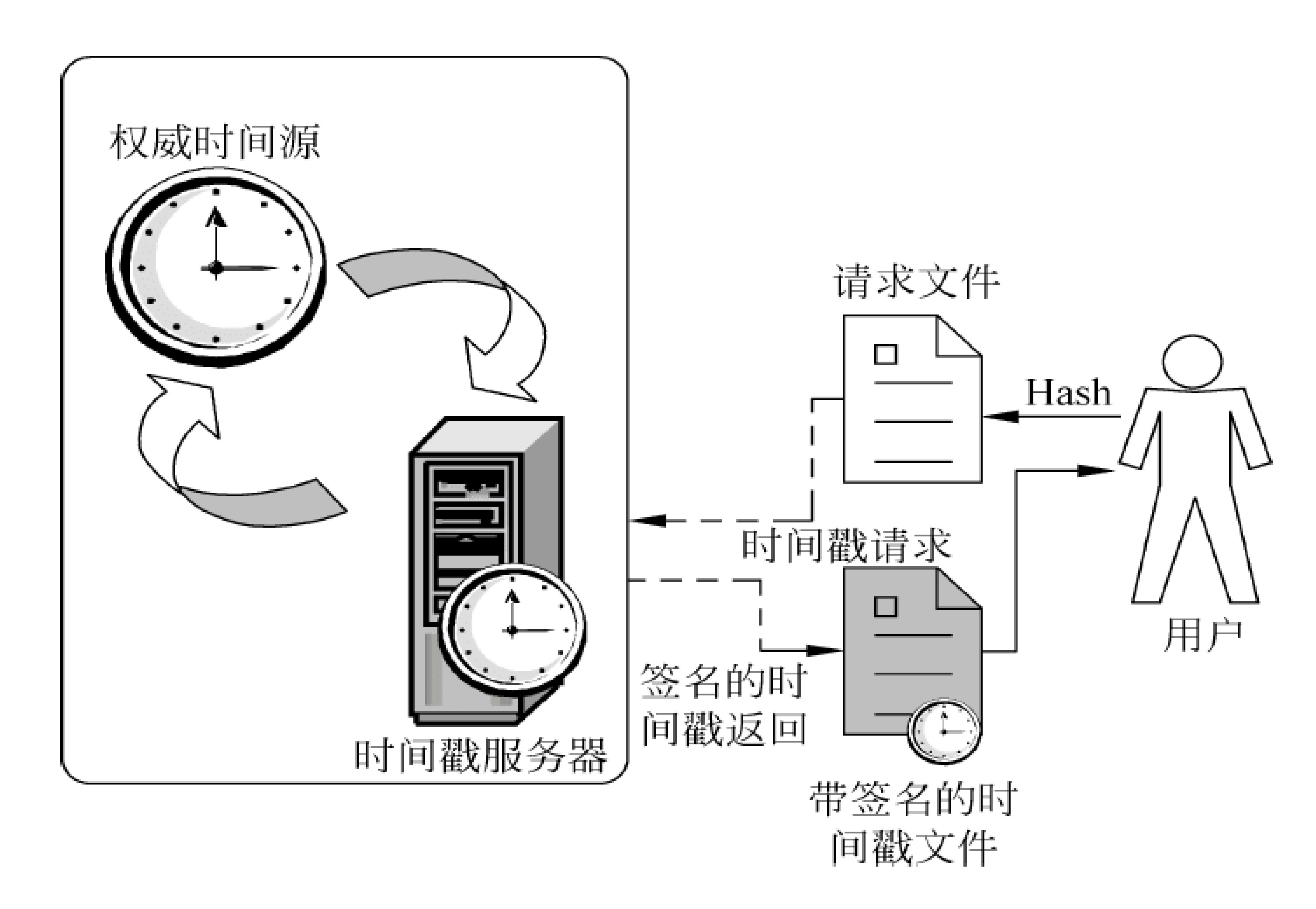



图 5-7　时间戳服务工作原理

在很多情况下，需要证明某个文件在某个时期存在。版权或专利争端就是谁拥有产生争议的工作的最早的副本，谁就将赢得官司。对于纸上的文件，公证人可以对文件签名，律师可以保护副本。如果产生了争端，公证人或律师可以证明某份文件产生于某个时间。

在数字世界中，事情要复杂得多。没有办法检查篡改签名的数字文件。它们可以被无止境地复制和修改而无人发现。在计算机文件上改变日期标记是轻而易举的事，没有人在看到数字文件后说："是的，这个文件是在 1952 年 12 月 4 日以前创建的。"

Bellcore 的 Stuart Haber 和 W. Scott Stornetta 研究了这个问题，认为数字时间标志协议有下列 3 条性质：

(1) 数据本身必须有时间标记，而不考虑它所用的物理媒介。

(2) 必定不存在改变文件的 1 个比特，而文件却没有明显变化。

(3) 必定不可能用不同于当前日期和时间的日期和时间来标记文件。

针对时间戳服务的实现原理及功能特点，其目前主要应用包括金融服务、个人或者公司银行服务、股票交易、零售业、直销、政府、在线拍卖、制造、供应链管理系统和医疗网络等领域。

#### 5.3.2.1　仲裁解决方法

这个协议的参与方是 T 和 A，其中 T 提供可信的时间标记服务，A 希望对文件加上时间标记：

(1) A 将文件的副本传送给 T。

(2) T 将他收到文件的日期和时间记录下来，并妥善保存文件的副本。

现在，如果有人对 A 所声明的文件产生的时间有怀疑，A 只要打电话给 T，T 将提供文件的副本，并证明他是在标记的日期和时间接收到文件，就可以排除对 A 的怀疑。

该解决方案如图 5-8 所示。

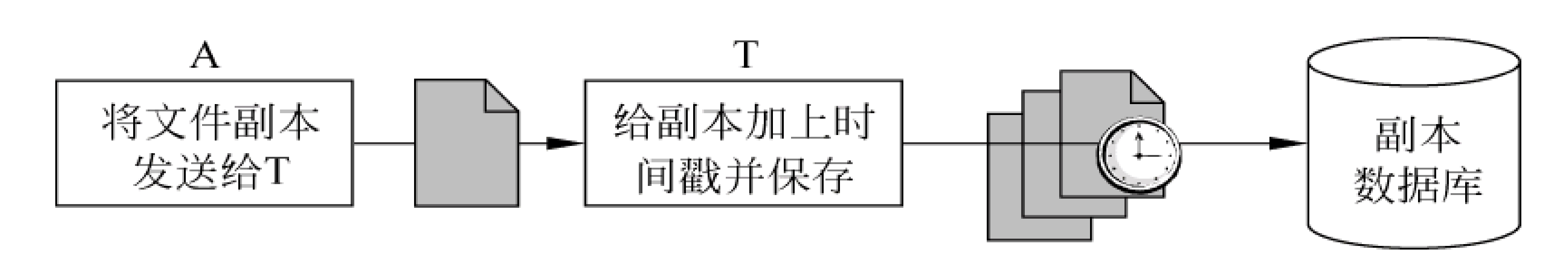



图 5-8 仲裁解决方法

这个协议是可行的，但有下列明显问题：

(1) 没有保密性，A 不得不将文件的副本交给 T。在信道上窃听的任何人都可以读它。虽然 A 可以对文件加密，但文件仍要放入 T 的数据库中，谁知道这个数据库有多安全？

(2) 数据库本身将是巨大的。并且发送大量的文件给 T 所要求的带宽也非常大。

(3) 存在潜在错误。传送错误或 T 的中央计算机中某些地方的电磁炸弹引爆将使 A 声明的时间标志完全无效。

(4) 可能有些运行时间标记业务的人并不像 T 那样诚实。也许没有任何事情能阻止 A 和 T 合谋，用他们想要的任何时间对文件作时间标记。

#### 5.3.2.2 改进的仲裁解决方法

单向 Hash 算法和数字签名能够轻而易举地解决上面所述的大部分问题，采用这种方式时，协议修改为：

(1) A 产生文件的单向 Hash 值。

(2) A 将 Hash 值传送给 T。

(3) T 将接收到 Hash 值的日期和时间附在 Hash 值后，并对结果进行数字签名。

(4) T 将签名的散列和时间标记送回给 A。

该改进方案如图 5-9 所示。

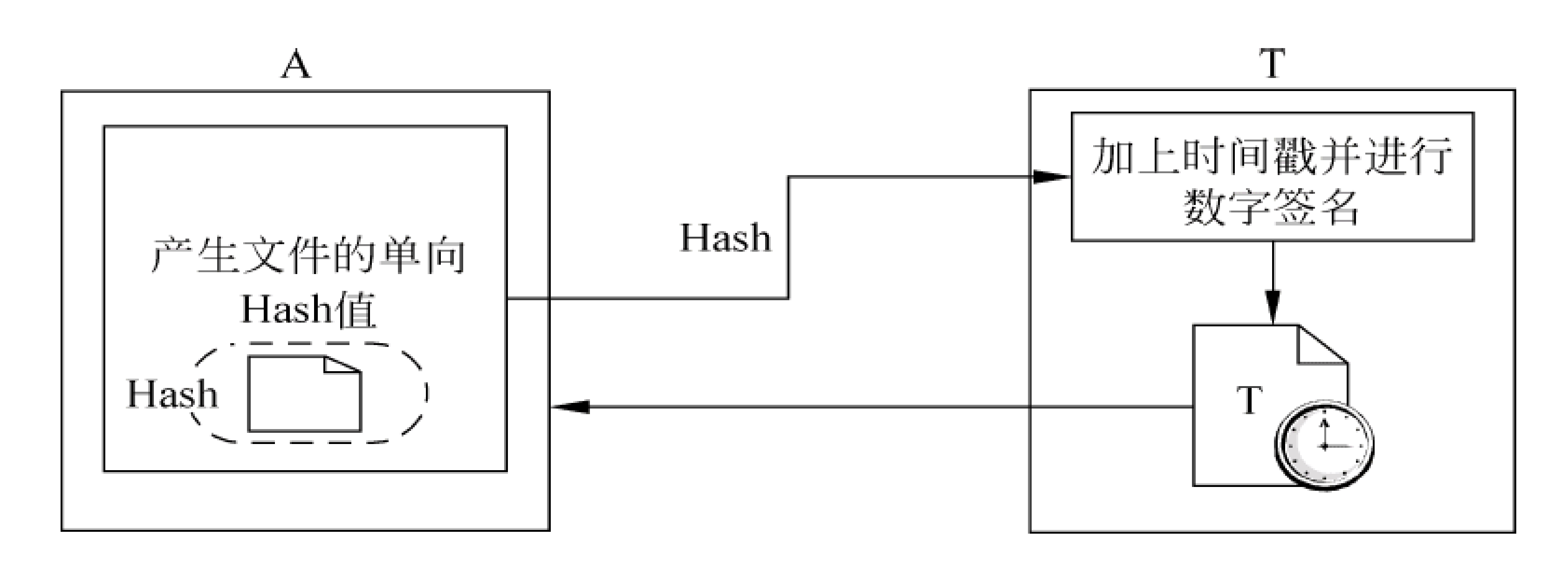



图 5-9 改进的仲裁解决方法

这种方法解决了 5.2 节提到的除最后一个问题外的所有问题。A 再也不用担心展示她的文件内容，因为 Hash 值就足够了。T 也不用存储文件的副本（或者 Hash 值）。这样，大量的存储要求和安全问题被解决了，因为单向 Hash 算法不需要密钥。A 可以立刻检查她在步骤(4)中接收到的对时间标记的 Hash 值的签名。这样，她将立刻发现在传送过程中是否存在错误。

这里还存在的问题是，A 和 T 仍然可以合谋来产生他们想要的任何时间标记。

#### 5.3.2.3　链接协议

解决这个问题的一种方法是将 A 的时间标记同以前由 T 产生的时间标记链接起来。这些时间标记很可能是为其他人产生而不是为 A 产生的。由于 T 预先不知道他所接收的不同时间标记的顺序，A 的时间标记一定发生在前一个时间标记之后。并且由于后面来的请求是与 A 的时间标记链接，那么她必须出现在前面。A 的请求正好夹在两个时间之间。

如果 A 想要做时间标记的 Hash 值是 $H_n$，并且前一个时间标记是 $T_{n-1}$，那么协议如下：

(1) A 将 $H_n$ 和 A 发送给 T。

$$H_n, A \rightarrow T$$

(2) T 将如下消息送回给 A。

$$T_n = S_k(n, A, H_n, T_n; I_{n-1}, H_{n-1}, T_{n-1}, L_n) \rightarrow A$$

这里，$L_n$ 是由下面的 Hash 链接信息组成。

$$L_n = H(I_{n-1}, H_{n-1}, T_{n-1}, L_{n-1})$$

$S_K$ 表示信息是用 T 的私钥签名的。A 的名字表明她是请求的发起者，参数 n 表示请求的序号：这是 T 发布的第 n 个时间标志。参数 $T_n$ 是时间。

另外的信息是标识 $I_{n-1}$、源 Hash 值 $H_{n-1}$、时间 $T_{n-1}$ 和 T 对以前文件做的时间标记的 Hash 值 $L_n$。

(3) 在 T 对下一个文件做时间标记后，他将那个文件发起者的标识符 $I_{n+1}$ 发送给 A。

如果有人对 A 的时间标记提出疑问，她只同自己前后文件的发起者 $I_{n-1}$ 和 $I_{n+1}$ 接触就行了。如果对她前后文件也有疑问，他们可以同 $I_{n-2}$ 和 $I_{n+2}$ 接触等，每个人都能表明他们的文件是在先来的文件之后和后来的文件之前打上时间标记的。

该协议过程如图 5-10 所示。

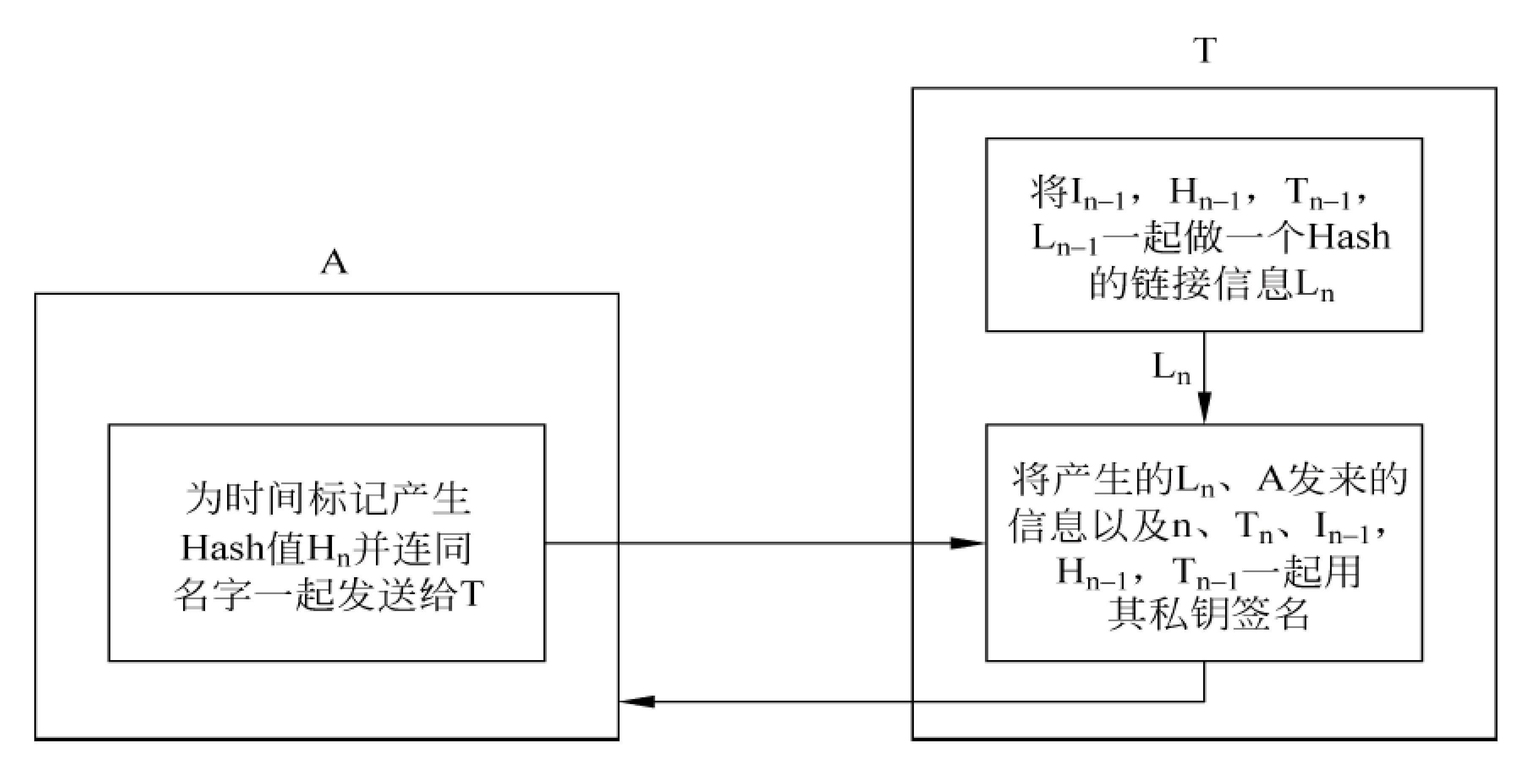



图 5-10　链接协议

这个协议使 A 和 T 很难合谋去产生一个文件的时间标记，使它不同于实际的时间标记。T 不可能为 A 顺填文件的日期。如果这样 T 就要预先知道在它之前是哪个文件的请求。即使他伪造那个文件，但也得知道在那个文件前来的是什么文件的请求等。由于时间标记必须嵌入到马上发布的后一个文件的时间标记中，并且那个文件也已经发布了，他不可能倒填文件的日期。破坏这个方案的唯一的办法是在 A 的文件前后创建一个虚构的文件链，该链足够长，从而穷举任何人对时间标记提出的疑问。

### 5.3.2.4 分布式协议

人去世之后，时间标记就会丢失。在时间标记和质询之间很多事情都可能发生，以使 A 不可能得到 $I_{n-1}$ 的时间标记的副本，这个问题可以通过把前面 10 个人的时间标记嵌入 A 的时间标记中得到缓解，并且将后面 10 个人的标识都发给 A。这样 A 就会有更大的机会找到那些仍有他们时间标记的人。

按照类似的方法，下面的协议与 T 一起实现分布式协议。

(1) 用 $H_n$ 作为输入，A 用密码安全的伪随机数发生器产生一串随机值。

$$V_1, V_2, V_3, \cdots, V_k$$

(2) A 将这些值的每一个看作其他人的 I 身份标识。她将 $H_n$ 发送给他们中的每个人。

(3) 他们中的每个人将日期和时间附到 Hash 值后，对结果签名，并将它送回给 A。

(4) A 收集并存储所有的签名作为时间标记。

协议过程如图 5-11 所示。

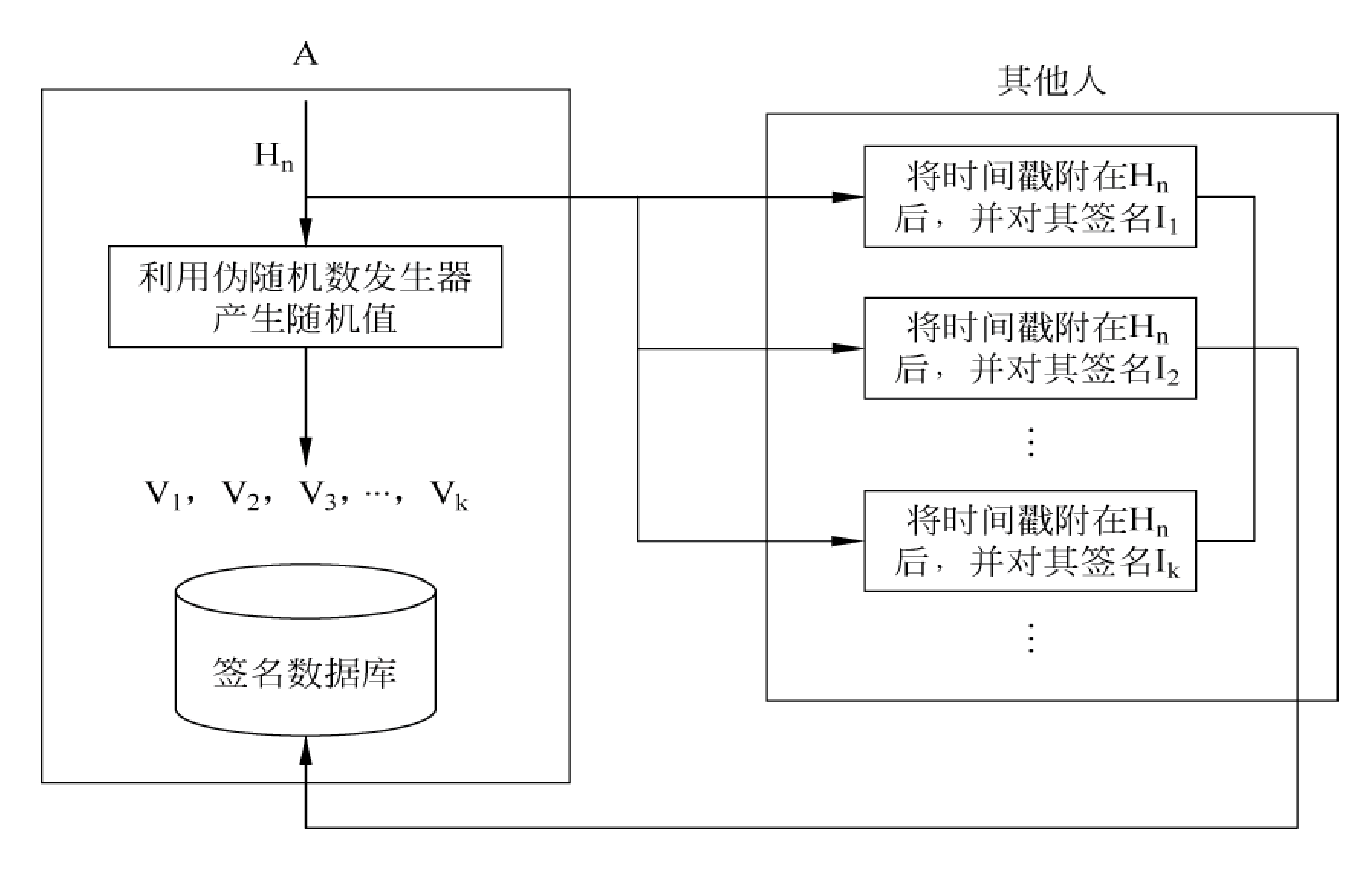



图 5-11 分布式协议

步骤(1)中密码安全的伪随机数发生器防止了 A 故意选取不可靠的 I 作为证人。即使她在自己的文件中作些改变以便构造一组不可靠的 I，她用这种方式逃脱的机会也是很小的，Hash 算法使 I 随机化了。A 不可能强迫他们。

这个协议是可行的，因为 A 伪造时间标记的唯一办法是使 K 个人都与她合作。由于她在步骤(1)中随机地选择 K 个人，因此防备这种攻击的可能性是很高的。社会越腐败，K 值就应越大。

另外，应该有一些机制来对那些不能马上返回时间标记的人进行处理。K 的一些子集都应是有效时间标记所要求的。其细节由具体的实现来决定。

#### 5.3.2.5　二叉树方法

时间标记协议的一种改进是利用二叉树来增加时间标记的数目。这个时间标记的数目依赖于一个给定的时间标记，以进一步减少某些人产生虚拟时间标记链的可能性。在公共地方也可以公布每天时间标记的 Hash 值，例如在报纸上。这类似于在分布协议中发送 Hash 值给随机的人。事实上，从 1992 年起时间标记就已经出现在每星期日的《纽约时报》上了。

这些时间标记协议取得了专利权。原隶属于 Bellcore 公司的 Surety 技术公司拥有这些专利，并将数字公证系统推向市场以支持这些协议。

在第一版中，客户发出“证明”请求给中央协调服务器。

下述 Merkle 技术使用 Hash 算法构造树：服务器构造由 Hash 值构成的树，树的叶子是在给定时间秒期间所有接收的请求，并且服务器把从它的叶子到树根的路径上 Hash 值的列表发回给每位请求者。客户软件把它存储在本地，并能为已经证明的任何文件发布一个数字公证的“证书”。这些树的根序列由在多个储存库地点用电子手段获得的“全程有效记录”组成(也在 CD-ROM 上发布)。客户软件也包括一个“有效”函数，允许用户测试文件是否已经被准确地用其当前形式证明(对适当的树根通过查询储存库，并把它与从文件和它的证书中重新计算出的 Hash 值进行比较)。

### 5.3.3　隐蔽信道通信的需求

假设 A 和 B 被捕入狱。A 将去男牢房，而 B 则进女牢房。看守 W 同意 A 和 B 交换消息，但不允许他们加密。因为 W 认为他们可能会商讨逃跑计划，因此，W 想能够阅读他们信息交换的每个细节。

W 也希望欺骗 A 和 B，他想让他们中的一个将一份欺诈的消息当作来自另一个的真实消息。A 和 B 愿意冒这种被欺骗的危险，否则他们根本无法联络，而他们必须商讨计划。

为了完成这件事，他们不得不欺骗看守，并找出一个秘密通信的方法。他们不得不建立一个隐蔽信道，即他们之间完全在 W 视野内的一个秘密通信信道，即使消息本身并不包含秘密信息。通过交换完全无害的签名消息，他们可以来回传送秘密信息，并骗过 W，即使 W 正在监视所有的通信。

一个简易的隐蔽信道可以是句子中单词的数目。句子中奇数个单词对应“1”，而偶数个单词对应“0”。因此，当读到这种仿佛无关的段落时，已将消息“101”送给了在现场的自己一方的人。这种技术的问题在于它仅仅是密码；没有密钥，安全性依赖于算法的保密性。

Gustavus Simmons 发明了传统数字签名算法中隐蔽信道的概念。由于隐蔽消息隐藏在看似正常数字签名的文本中,这是一种迷惑人的形式。W 看到来回传递的签名的无害消息,但他完全看不到通过隐蔽信道传递的信息。事实上,隐蔽信道签名算法与通常的签名算法不能区别,至少对 W 是这样。W 不仅不能读隐蔽信道消息,而且他也不知道隐蔽信道消息已经出现。

一般地,协议过程如下:

(1) A 产生一个无害消息,最好是随机的。

(2) 用与 B 共享的密钥,A 对这个无害信息这样签名,她在签名中隐藏她的隐蔽信息。

(3) A 通过 W 发送签名消息给 B。

(4) W 读这份无害的消息并检查签名,没发现什么问题,他将这份签了名的消息传递给 B。

(5) B 检查这份无害消息的签名,确认消息来自于 A。

(6) B 忽略无害的消息,而用他与 A 共享的密钥,提取隐蔽消息。

该协议如图 5-12 所示。

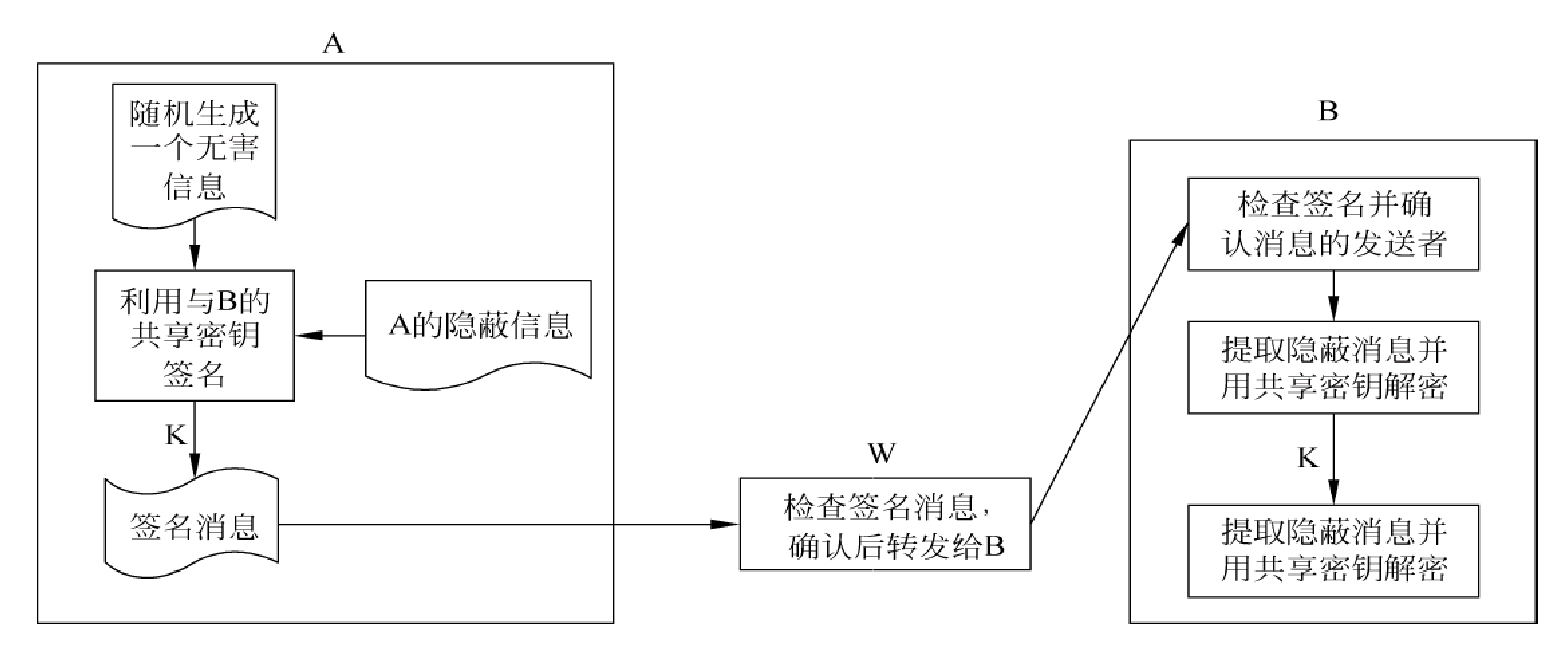



图 5-12 隐蔽信道

怎样进行欺骗呢? W 不相信任何人,别人也不相信他。他是可以阻止通信,但他没法构造虚假信息。如果他构造假信息,由于他没法产生任何有效的签名,B 将在步骤(5)中检测出他的意图。并且,由于他不知道共享密钥,他没法阅读隐蔽信息。更重要的是,他不知道隐蔽信息在哪里。用数字签名算法签名后的消息与嵌入到签名中的隐蔽消息看上去没有不同。

A 和 B 之间的欺骗问题就更多。在隐蔽信道的一些实现中,B 需要从隐蔽信道读的秘密信息与 A 需要签名的无害信息是相同的。如果这样,B 能够冒充 A。他能对消息签名而声称该消息来源于她,而对此 A 无能为力。如果她要给他发送隐蔽消息,她不得不相信他不会滥用她的私钥。

别的隐蔽信道应用中没有这个问题。由 A 和 B 共享的密钥允许 A 给 B 发送隐蔽信

息，但这个密钥与 A 的私钥不同，并不允许 B 对消息签名。A 也就不必相信 B 不会滥用她的私钥了。

#### 5.3.3.1　隐蔽信道的应用

隐蔽信道最显著的应用是在间谍网中。如果每人都收发签名消息，间谍在签名文件中发送隐蔽信息就不会被注意到。当然，敌方的间谍也可以做同样的事。

用一个隐蔽信道，A 可以在受到威胁时安全地对文件签名。她可以在签名文件时嵌入隐蔽消息，说“我被胁迫”。

另外的应用则更为微妙：公司可以签名文件，嵌入隐蔽信息，允许它们在整个文档有效期内被跟踪。政府可以“标记”数字货币。恶意的签名程序可能泄露其签名中的秘密信息等，其可能性是无穷的。

#### 5.3.3.2　杜绝隐蔽的签名

A 和 B 互相发送签名消息，协商合同的条款，使用数字签名协议。但是，这个合同谈判是用来掩护 A 和 B 的间谍活动。当他们使用数字签名算法时，他们不关心所签名的消息。他们利用签名中的隐蔽信道彼此传送秘密信息。

然而，反间谍机构不知道合同谈判以及签名消息的应用只是表面现象。因此人们创立了杜绝隐蔽的签名方案。这些数字签名方案不能被变更让其包含隐蔽信道。

### 5.3.4　不可抵赖的数字签名

一般的数字签名可以被准确复制。这个性质有时是有用的，比如公开宣传品的发布。在其他情况下，它则可能有问题。

想像一下数字签名的私人或商业信件。如果到处散布那个文件的许多复制，而每个复制又能够被任何人验证，这样可能会导致窘迫或勒索。最好的解决方案是数字签名能够被证明是有效的，但没有签名者的同意，接收者不能把它给第三方看。

举一个例子：A 软件公司发布 DEW 软件(do-everything-word)。为了确认软件中不带病毒，他们在每个复制中包括一个数字签名。但是，他们只想让软件的合法买主能够验证数字签名，盗版者则不能。同时，如果 DEW 复制中发现有病毒，A 软件公司不能否认一个有效的数字签名。

不可抵赖签名适合于这类任务。与通常的数字签名类似的是，不可抵赖签名依赖于签名的文件和签名者的私钥。但不同的是，如果没有得到签名者同意，不可抵赖签名就不能验证。

虽然对这些签名，用像“不可改变的签名”一类的名称更好，但这个名称的由来是如果 A 被强迫承认或抵赖一个签名(很可能在法庭上)，她不可能不实地否认她的真实签名。

数学描述是复杂的，但其基本思想是简单的：

(1) A 向 B 出示一个签名。

(2) B 产生一个随机数并送给 A。

(3) A 对收到的随机数利用自己的私钥进行计算,将计算结果送给 B。

(4) A 只能计算该签名是否有效。

B 可以确认这个结果。

上述思想如图 5-13 所示。

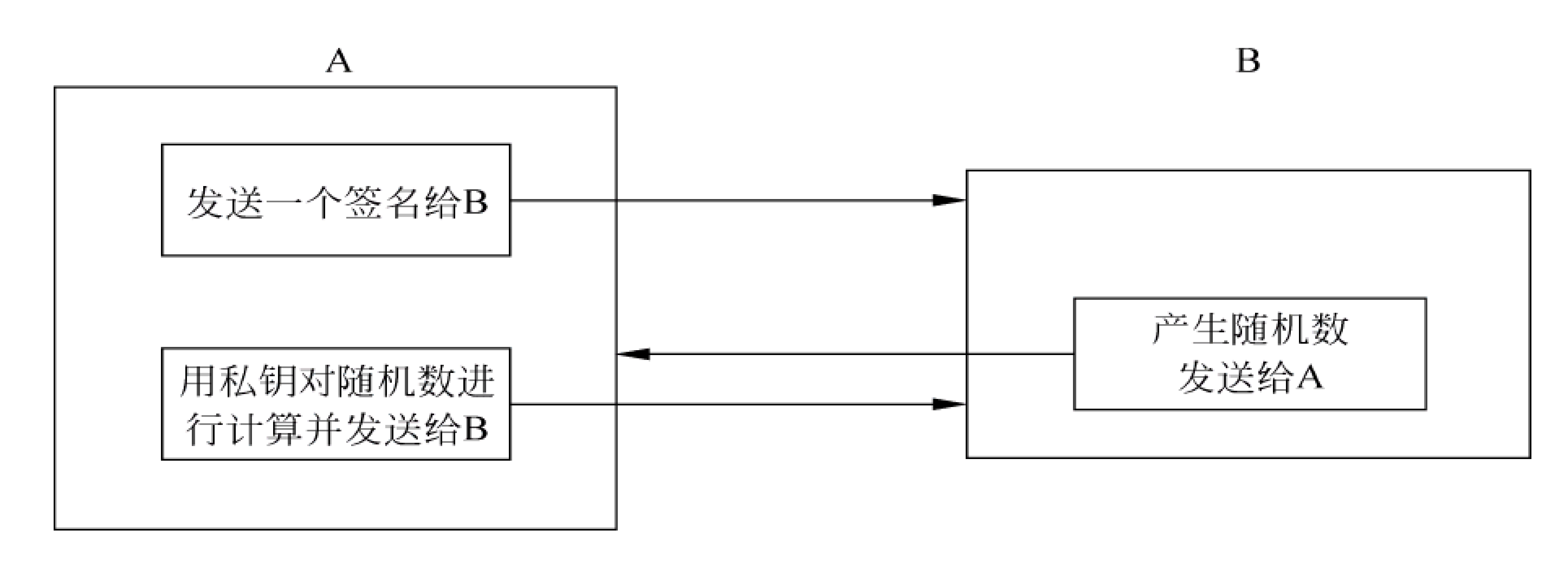



图 5-13 不可抵赖的数字签名

也有另外的协议,使 A 能够证明她没有对文件签名,同时又不能抵赖签名。

B 不能转而让 C 相信 A 的签名是有效的,因为 C 不知道 B 的数字是随机数,而且 B 很容易在文件上完成这个协议,而不用 A 的任何帮助,然后就将结果出示给 C。所以,C 只有在她与 A 本人完成这个协议后,才能相信 A 的签名是有效的。

这个解决方法并不完美,Yvo Desmedt 和 Moti Yung 的研究表明,在某些情况下,B 让 C 相信 A 的签名有效是可能的。

例如,B 买了 DEW 的一个合法复制,他能在任何时候验证软件包的签名。然后,B 使 C 相信他是来自于 A 软件公司的销售商。他卖给她一个 DEW 的盗版。当 C 试图验证 B 的签名时,他同时要验证 A 的签名。当 C 发给 B 随机数时,B 把它送给 A;当 A 响应后,B 就将响应送给 C。于是 C 相信她是该软件的合法买主,尽管她并不是。

即使如此,不可抵赖的签名仍有许多应用,在很多情况中,A 不想任何人都能够验证她的签名。她不希望自己的个人通信被媒体核查、展示或者在事情已经改变后被验证。如果她对自己卖出的信息签名,她不希望没有对信息付钱的那些人能够验证它的真实性。控制谁验证她的签名是 A 保护她的个人隐私的一种方法。

不可抵赖签名的一种变化是把签名者与消息之间的关系与签名者与签名之间的关系分开。在这种签名方案中,任何人能够验证签名者实际产生的签名,但签名者的合作者还需要验证该消息的签名是有效的。

相关的概念是"受托不可抵赖签名"。设想 A 为某公司工作,并使用不可抵赖的签名协议发送控告文件给报纸。A 能够对报社记者验证她的签名,但不向其他任何人验证签名。然而执行总裁 B 怀疑文件是 A 提供的,他要求 A 执行否认协议来澄清她的名字,A 拒绝了。B 认为 A 不得不拒绝的唯一理由是她有罪,于是便解雇她。

除了否认协议只能由 T 执行外,受托不可抵赖签名类似不可抵赖签名。B 不能要求 A 执行否认协议,只有 T 能够。如果 T 是法院系统,那么他将只在解决正式的争端的时候执行协议。

## 5.3.5　指定的确认者签名

A 公司销售 DEW 软件的生意非常兴隆，但是，A 花在验证不可抵赖签名的时间比编写新功能部件的时间更多。

所以，A 希望有一种办法可以在公司中指定一个特殊的人，比如 C，负责对整个公司的签名验证。A 或任何其他程序员可以用不可抵赖协议对文件签名，但是所有验证都由 C 处理。

研究表明，用指定的确认人签名是可行的。A 可以对文件签名，而 B 相信签名是有效的，但他不能使第三方相信。同时，A 可以指定 C 作为她签名后的确认人。A 甚至事先不需要得到 C 的同意，她只需要 C 的公开密钥。如果 A 不在家，已经离开公司，或者突然死亡了，C 仍然能够验证 A 的签名。

指定的确认人签名是标准的数字签名和不可抵赖签名的折中。肯定有一些场合 A 可能想要限制能验证她的签名的人。另一方面，如果 A 完全控制，则破坏了签名的可实施性：A 可能在确认或否认方面拒绝合作，她可能声称用于确认或否认的密钥丢失了，或者她可能正好身份不明。指定的确认人签名让 A 既能保护不可抵赖签名同时又不让她滥用这种保护。A 可能更喜欢那种方式：指定的确认人签名能够帮助她防止错误的应用，如果她确实丢失了密钥，可以保护她，如果她是在度假、在医院，甚至死了，也可以插手干预。

这种想法有各种可能的应用。C 能够把她自己作为公证人公开。她能够在一些地方的一些目录中发布她的公开密钥，人们能够指定她作为他们签名的确认人。她向大众收取少量的签名确认费用，使她可以生活得很好。

C 可能是版权事务所、政府机构、其他的很多事物。这个协议允许组织机构把签署文件的人同帮助验证签名的人分开。

## 5.3.6　代理签名

指定确认人签名允许签名者指定其他某个人来验证他的签名。但是，如果 A 需要到一些地方进行商业旅行，这些地方不能很好地访问网络（例如到非洲丛林）。或者她在大手术后，无能为力。她希望接收一些重要的电子邮件，并指示她的秘书 B 作相应的回信。A 在不把她的私钥给 B 的情况下，该如何让 B 行使她的消息签名权利呢？

代理签名是一种解决方案。A 可以给 B 代理，这种代理具有下面的特性：

（1）可区别性。任何人都可区别代理签名和正常签名。

（2）不可伪造性。只有原始签名者和指定代理签名者能够产生有效的代理签名。

（3）代理签名者的不符合性。代理签名者必须创建一个能检测到是代理签名的有效代理签名。

（4）可验证性。从代理签名中，验证者能够相信原始签名者认同了这份签名消息。

（5）可识别性。原始签名者能够从代理签名中识别代理签名者的身份。

（6）不可抵赖性。代理签名者不能否认他创立的且被认可的代理签名。

在某些情况下，需要更强的可识别性形式，即任何人都能从代理签名中确定代理签名

者的身份。

### 5.3.7 团体签名

David Chaum 提出了下述问题：一个公司有几台计算机，每台都连在局域网上。公司的每个部门有自己的打印机，也连在局域网上，并且只有本部门的人员可以使用自己部门的打印机。因此，打印前，必须使打印机确认用户是在哪个部门工作的。同时，公司想保密，不可以暴露用户的姓名。

然而，如果在当天工作结束时发现打印机用得太频繁，主管者必须能够找出谁滥用了那台打印机。

对这个问题的解决方法称为团体签名。它具有以下特性：

(1) 只有该团体内的成员能对消息签名。

(2) 签名的接收者能够证实消息是该团体的有效签名。

(3) 签名的接收者不能决定是该团体内哪一个成员签的名。

(4) 在出现争议时，签名能够“打开”，以揭示签名者的身份。

下面采用具有可信仲裁者的团体签名来解决这一问题。利用可信仲裁者 T：

(1) T 生成一大批公开密钥/私有密钥对，并且给团体内每个成员一个不同的唯一私钥表。在任何表中密钥都是不同的(如果团体内有 n 个成员，每个成员得到 m 个密钥对，那么总共有 $n \times m$ 个密钥对)。

(2) T 以随机顺序公开该团体所用的公开密钥主表。T 保存一个哪些密钥属于谁的秘密记录。

(3) 当团体内成员想对一个文件签名时，他从自己的密钥表中随机选取一个密钥。

(4) 当有人想验证签名是否属于该团体时，只需查找对应公钥主表并验证签名。

(5) 当争议发生时，T 知道哪个公钥对应于哪个成员。

该协议如图 5-14 所示。

这个协议的问题在于需要可信的一方。T 知道每个人的私钥因而能够伪造签名。而且，m 必须足够长以避免试图分析出每个成员用的哪些密钥。

### 5.3.8 失败-终止数字签名

假想 E 是非常强的敌人，拥有巨大的网络和很多装满了超级计算机的屋子，其计算机的能力比 A 大许多级。这些计算机昼夜工作试图破译出 A 的私钥，最终成功了。E 现在就能够冒充 A，随意地在文件上伪造她的签名。

由 Birgit Pfitzmann 和 Michael Waidner 引入的失败-终止数字签名可以避免这种欺诈。如果 E 在穷举攻击后伪造 A 的签名，那么 A 能够证明它们都是伪造的。如果 A 对文件签名，然后否认签名，声称是伪造的，法院能够验证它不是伪造的。

失败-终止签名的基本原理是：对每个可能的公开密钥，都对应有多种可能的私钥和它一起工作。这些私钥中的每一个产生许多不同的可能的签名。然而，A 只有一个私钥，只能计算一个签名。A 并不知道别的任何私钥。

E 试图破解出 A 的密钥。E 也可能是 A，试图为她自己计算第二个私钥。她收集签

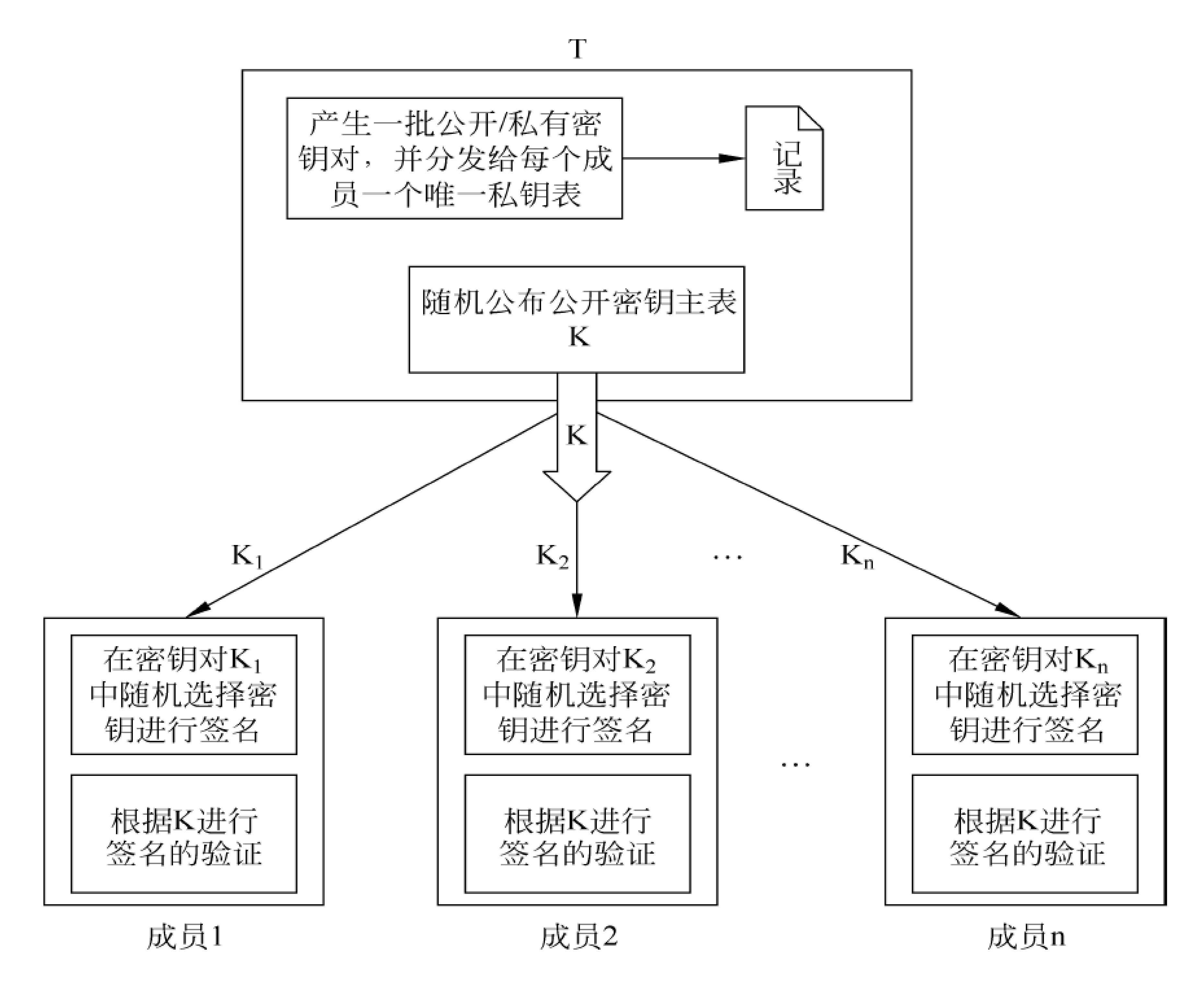



图 5-14　具有可信仲裁者的团体签名

名消息,并且利用她的超级计算机阵列,试图恢复出 A 的私钥。假设 E 能够恢复出有效的私钥,但因为有许多可能的私钥,因而正在使用的这个私钥可能是不同的一个。E 恢复出正确的私钥的概率非常小,可以忽略不计。

现在,当 E 利用她产生的私钥伪造签名时,它将不同于 A 本人对文件的签名。当 A 被传到法院,对同一消息和公开密钥,她可以生成另一个不同的签名,有别于 E 产生的签名,以证明是伪造的。而另一方面,如果 A 不能产生出不同的签名,就表明不存在伪造,A 就要对她的签名负责。

这个签名方案避免了 E 通过巨大的计算能力来破译 A 的签名方案。但它对下面这种更有可能发生的攻击却无能为力:当 M 闯入 A 的住宅并偷窃她的私钥或者 A 签署了一个文件然后却丢失了她的私钥时,为了防止这种攻击,A 应该自己负责,这种事情已超出了密码学的范围。

## 5.3.9　用加密的方法计算数据

设想如下情况,A 想知道某个函数 f(x)对某些特殊的 x 值的解。但是,她的计算机坏了,B 愿意为她计算 f(x),但 A 又不想让 B 知道她的 x。怎样做 A 才能在不让 B 知道 x 的情况下为她计算 f(x)呢?

这是加密数据计算的一般问题,亦称“对先知隐藏信息”问题。B 是先知,他回答问题。

对某些函数来说有办法解决这个问题。例如下面的情形:

已知一个大素数 p 和一个生成元 g，A 有 x，希望求得 e，使得

$$g^e \equiv x \pmod p$$

这种情形下，A 就可以在不泄漏 x 的情况下要求 B 帮助计算结果，协议过程如下：

(1) A 选择一个小于 p 的随机数 r。

(2) A 计算

$$x' = xg^r \bmod p$$

(3) A 要求 B 计算 e′，使得

$$g^{e'} \equiv x' \bmod p$$

(4) B 将计算的 e′发给 A。

(5) A 计算 $e=(e'-r) \bmod (p-1)$，求得 e。

## 5.3.10 公平的硬币抛掷的游戏和应用

有一种游戏：A 和 B 想抛掷一个公平的硬币，但又没有实际的物理硬币可用。A 提出一个用思维来抛掷公平硬币的方法。

“首先，你想一个随机比特，然后我再想一个随机比特。然后我们将这两个比特进行异或。”A 建议。

“但如果我们中有人不随机选比特怎么办呢?”B 问道。

“这无关紧要。只要这些比特中的一个是真正随机的，它们之异或应该也是真正随机的。”A 回答。

经过思考 B 同意了。

没过多久，A 和 B 碰到一本被丢弃在路旁的书。A 说：“我们中有一个必须拣起这本书，并找到一个合适的垃圾箱”。B 同意并提议用前面提到的抛币协议来决定谁必须将这本书扔掉。

“如果最后的比特是‘0’，那么你必须拣起那本书；如果是‘1’，那我必须那样做。”A 说。“你的比特是什么?”

B 答道：“1”。

“我的也是 1”，A 顽皮地说：“我猜想今天不是你的幸运日。”

很明显，这个抛币协议有严重缺陷。虽然，真正随机的比特 x 与任意独立分配的比特 y 异或仍得到真正随机的比特。但是 A 的协议不能保证两个比特是独立分布的。

事实上，容易验证不存在能让两个能力无限的团体进行公平抛币的思维协议。

接下来，A 和 B 对原协议版本进行了修改。

首先，B 确定一个比特，但这次他不立即宣布，只是将它写在纸上，并装入信封中。接下来，A 公布她选的比特。最后，A 和 B 从信封中取出 B 的比特并计算随机比特。只要至少一方诚实地执行协议，这个比特的确是真正随机的。

A 和 B 有了这个可以工作的协议。

一般地，需要一个具有如下性质的协议：

(1) A 必须在 B 猜测之前抛币。

(2) 在听到 B 的猜测结果后，A 不能再抛币。

(3) B 在猜测之前必须不能知道硬币怎么落地的。

有以下几种方法可用来实现具有这些性质的协议。

#### 5.3.10.1　采用单向函数的抛币协议

如果 A 和 B 对使用一个单向函数达成一致意见,协议非常简单:

(1) A 选择一个随机数 x,她计算 y=f(x),这里 f(x)是单向函数。

(2) A 将 y 送给 B。

(3) B 猜测 x 是偶数或奇数,并将猜测结果发给 A。

(4) 如果 B 的猜测正确,抛币结果为正面;如果 B 的猜测错误,则抛币的结果为反面。A 公布此次抛币的结果,并将 x 发送给 B。

(5) B 确认 y=f(x)。

该协议如图 5-15 所示。

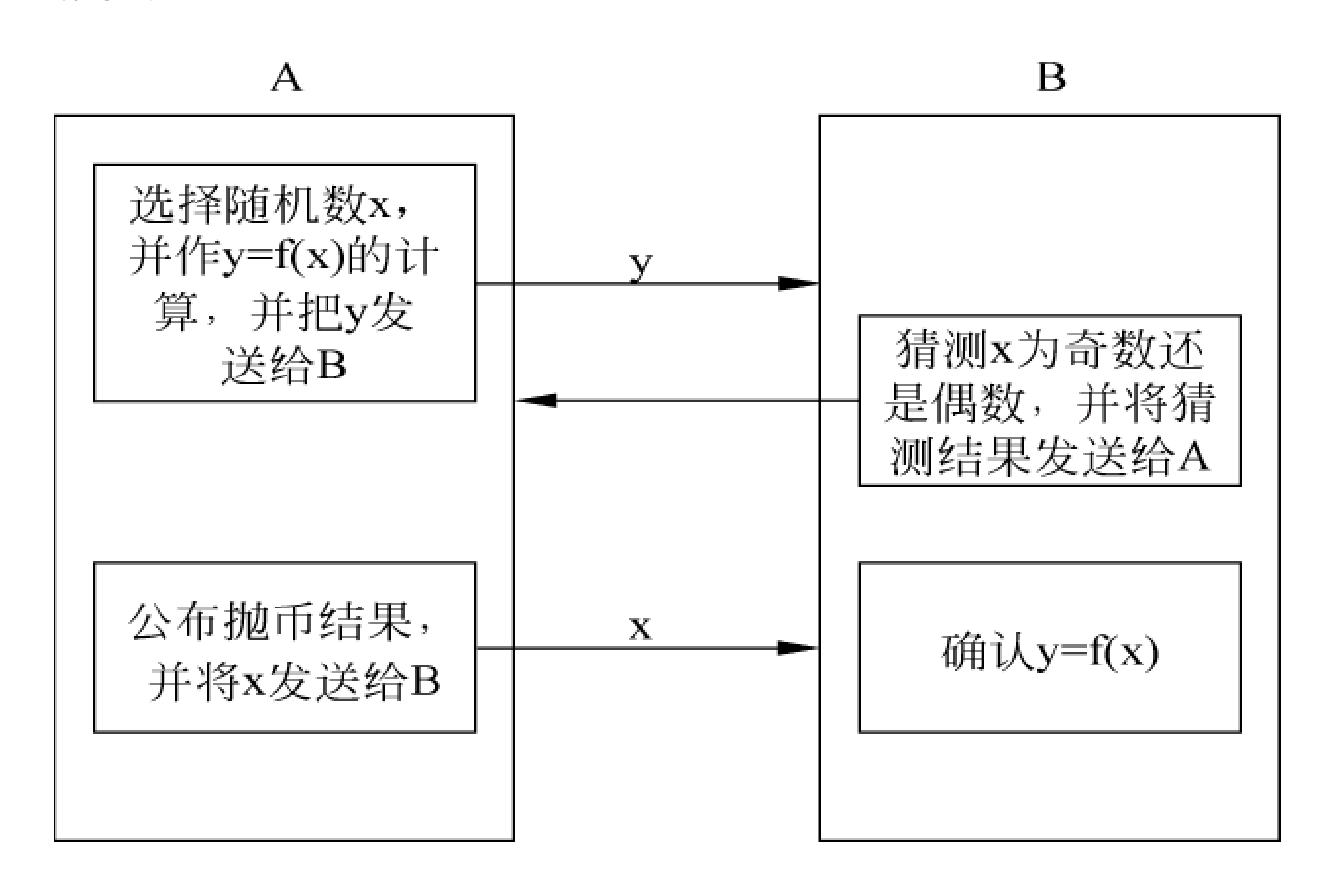



图 5-15　采用单向函数的抛币协议

此协议的安全性取决于单向函数。如果 A 能找到 x 和 x′,满足 x 为偶而满足 x′为奇,且 y=f(x)=f(x′),那么她每次都能欺骗 B。

f(x)没有意义的比特也必须与 x 不相关。否则,B 至少某些时候能够欺骗 A。例如,如果 x 是偶数,f(x)产生偶数的次数占 75%,B 就有优势。

#### 5.3.10.2　采用公开密钥密码的抛币协议

这个协议既可与公开密钥密码又可与对称密码一起工作。其唯一要求是算法满足交换律,即

$$D_{k_1}(E_{k_2}(E_{k_1}(M))) = E_{k_2}(M)$$

一般地,对称算法中这个特性并不满足,但对某些公开密钥算法是正确的(例如有相同模数的 RSA 算法)。协议如下:

(1) A 和 B 都产生一个公开/私有密钥对。

(2) A 产生两个消息,其一指示正面,另一个指示反面。这些消息中包含有某个唯一的随机串,以便以后能够验证其在协议中的真实性。A 用她的公开密钥将两个消息加

密,并以随机的顺序把他们发给 B,即:

$$E_A(M_1), E_A(M_2) \rightarrow B$$

(3) B 由于不能读懂其中任意一个消息,他随机地选择一个 M(M 是 $M_1$ 或 $M_2$)。再用他的公开密钥加密并回送给 A,即:

$$E_B(E_A(M)) \rightarrow A$$

(4) A 由于不能读懂送回给她的消息,就用她的私钥解密并回送给 B,即:

$$D_A(E_B(E_A(M))) = E_B(M_1),$$

如果

$$M = M_1;$$

或

$$D_A(E_B(E_A(M))) = E_B(M_2),$$

如果

$$M = M_2$$

(5) B 用他的私钥解密消息,得到抛币结果。他将解密后的消息送给 A:

$$D_B(E_B(M_1)) = M_1 \rightarrow A; \text{或} D_B(E_B(M_2)) = M_2 \rightarrow A$$

(6) A 读抛币结果,并验证随机串的正确性。

(7) A 和 B 出示他们的密钥对以便双方能验证对方没有欺诈。

该协议过程如图 5-16 所示。

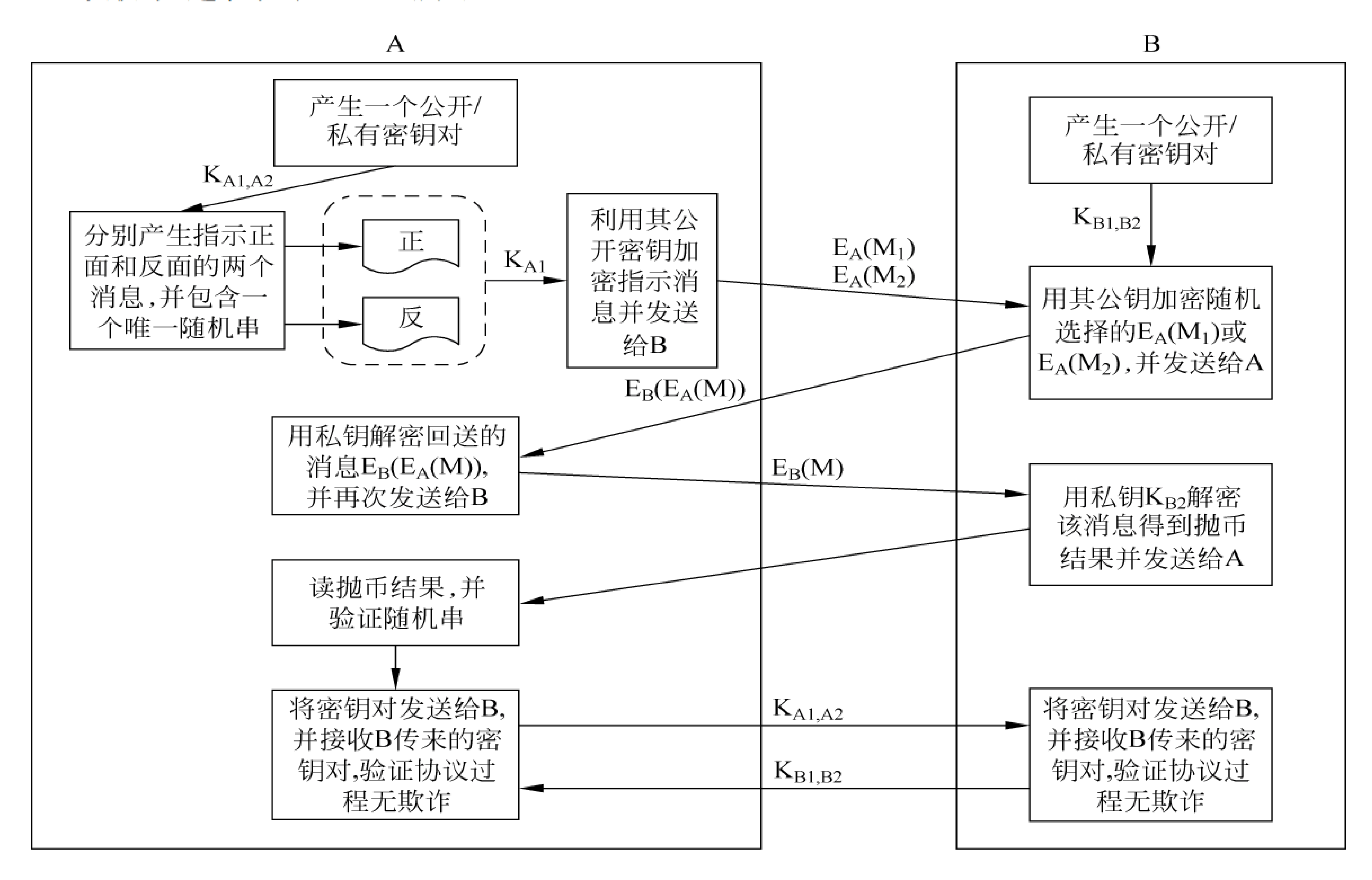



图 5-16　采用公开密钥密码的抛币协议

这个协议是自我实施的。任意一方都能即时检测对方是否有欺诈行为,不需要可信的第三方介入实际的协议和协议完成后的任何仲裁。下面考虑有欺诈的情形,看看协议是如何工作的。

如果 A 想欺骗,强制为正面,她有 3 种可能的方法影响结果。

第一种方法:她可以在步骤(2)中加密两个“正面”的消息。但在步骤(7)中 A 出示她的密钥时,B 就可以发现这种欺骗。

第二种方法:A 在步骤(4)时用一些其他的密钥解密消息,将产生一些乱七八糟的无用信息,而 B 可在步骤(5)中发现。

第三种方法:A 在步骤(6)中否认消息的有效性,但当在步骤(7)中 A 不能证明消息无效时,B 就可以发现。

当然,A 可以在任何一步拒绝参与协议,那样,A 欺骗 B 的企图就显而易见了。

如果 B 想欺骗并强制为“反面”,他的选择有以下几种。

(1) B 可以在步骤(3)中不正确地加密一个消息,但 A 在步骤(6)查看最终消息时就可以发现它。

(2) B 可以在步骤(5)中进行不适当的操作,但这也会导致乱七八糟的无用信息,A 可在步骤(6)中发现。

(3) B 可以声称由于 A 方的欺诈使他不能适当地完成步骤(5)的操作,但这种形式的欺诈能在步骤(7)中发现。

(4) B 可能在步骤(5)中给 A 一个“反面”的信息,而不管他解密获得的信息是什么,但 A 能在步骤(6)中立即检查消息的真实性。

#### 5.3.10.3　抛币的应用——会话密钥生成

抛币协议的一种实际应用是会话密钥生成。

掷币协议可以让 A 和 B 产生随机会话密钥,而且双方都不能影响密钥生成的结果。假定 A 和 B 加密他们交换的密钥,则这种密钥生成方法在存在窃听时也是安全的。

### 5.3.11　单向累加器

假设 A 是某公司的成员。有时候,她必须在光线暗淡的旅馆与其他成员会晤。由于旅馆的光线非常暗,以至于她难以知道桌子对面的人是否也是他们的成员。

该公司可以选择几种解决方案。每个成员可以携带一个成员名单,但这有两个问题:一是现在每人必须携带一个大的数据库,二是他们必须很好地保护成员名单。另一种选择是,通过一个值得信任的秘书发布数字签名的身份卡。这样做增加了让外来者验证成员的好处。但是它需要可信任的秘书,在该公司没有人能够被信任到那种程度。

新的解决方案是使用叫做“单向累加器”的东西。除了可交换外,它类似单向 Hash 算法。也就是说,用任何顺序对成员数据库进行 Hash 运算,都可以得到相同的值。而且,把新成员进行 Hash 运算,得到的新 Hash 值,也是与顺序无关的。

假定 n 是两个素数的乘积,与 $x_0$ 一起是事先约定的,则一种单向累加器函数如下:

$$A(x_i, y) = x_{i-1}^{y} \bmod n$$

成员 $y_1, y_2, y_3$ 的累加是：

$$((x_0^{y1} \bmod n)x^{y2} \bmod n)x^{y3} \bmod n$$

这个计算的结果与 $y_1, y_2, y_3$ 的顺序无关。

下面是 A 要做的事情。她计算除她自己外的每个成员名字的累加和。然后，她把这个值与她的名字存在一起。B 和其他每个成员都做和 A 类似的事。

现在，当 A 和 B 在光线暗淡的旅馆会面时，他们简单的互相交换累加值和名字，A 确认 B 的名字加上他的累加值等于 A 的名字加上她的累加值。B 也确认同样的事情。现在他们两人知道彼此是公司成员。同时，没有人能够知道任何其他人的身份。

甚至更好，非会员能知道每个人的累加值。现在 A 能够对非会员验证他的会员资格（也许，在他们的本地反间谍商店为会员打折），非会员不可能计算出全部会员资格的名单。

只要到处发送新会员的名字就可把新会员加入到累加值中。不幸的是，删除会员的唯一方法是给每个会员发送新名单并让他们重新计算他们的累加值。但是如果有人辞职，该公司就需要那样做；而死亡的会员可以保留在名单上。

在没有集中签名者的情况下，无论什么时候想要与数字签名有同样的效果时这是一个很好的想法，并且已获得应用。

### 5.3.12　秘密的全泄露或无泄露

假设 A 是前苏联的前代理商，现在失业了。A 为了挣钱，便出卖机密，任何愿意付钱的人都可以买秘密。A 罗列一个目录，所有秘密都编号列出，并加上一个非常撩人的标题，例如："谁在秘密控制着三方委员会？"、"为什么鲍里斯·叶利钦总是看上去像吞了一只活青蛙？"等。

A 不愿为一个秘密的价格而泄露两个秘密，或者泄露秘密的任何一部分信息。B 是一个潜在的买主，他不想为价值不高的秘密付钱，他也不想告诉 A 他想要哪个秘密。A 可能在她的目录中加上"B 对什么感兴趣"这一条。

在这种情况下，A 和 B 最后必须互相摊牌。当然，B 也可以进行欺骗从而获得不止一个秘密。

这个解决方案就叫做秘密的全或无泄露（all or nothing disclosure of secret, ANDOS）。因为，一旦 B 得到了不管是 A 的秘密中哪一个信息，他就失去了获知任何其他秘密的机会。协议如下：

首先，定义 FBI(fixed bit index)：两个字符 x，y 的 FBI 是 x 的第 i 位等于 y 的第 i 位的那些位，从右到左计，起始为 0，例如：

$$x = 110101001011$$
$$y = 101010000110$$
$$FBI(x, y) = \{1, 4, 5, 11\}$$

假定 A 是卖方，B 和 C 是买方。A 有 k 个 n 位的秘密：$S_1, S_2, \cdots, S_k$。B 想买秘密 $S_b$，C 想买秘密 $S_c$。

(1) A 产生一对公开/密钥对，并把公开密钥告诉 B，但不告诉 C；然后再产生另一对公开/密钥对，并把公开密钥告诉 C，但不告诉 B。

(2) B 产生 k 个 n 位随机数 $B_1, B_2, \cdots, B_k$ 并把结果告诉 C；C 产生 k 个 n 位随机数 $C_1, C_2, \cdots, C_k$，并把结果告诉 B。

(3) B 用从 A 得到的公开密钥加密 $C_b$，计算 $C_b$ 和加密结果的 FBI，并把 FBI 发给 C。

C 用从 A 得到的公开密钥加密 $B_c$，计算 $B_c$ 和加密结果的 FBI，并把 FBI 发给 B。

(4) B 对 n 位数 $B_1, B_2, \cdots, B_k$ 中的每一个数，凡是其索引不在他从 C 处接收到的 FBI 中出现的每一位，都用其代替，得到 $B'_1, B'_2, \cdots, B'_k$，并把它们发给 A。

C 对 n 位数 $C_1, C_2, \cdots, C_k$ 中的每一个数，凡是其索引不在他从 B 处接收到的 FBI 中出现的每一位，都用其代替，得到 $C'_1, C'_2, \cdots, C'_k$，并把它们发给 A。

(5) A 用 B 的密钥解密 $C'_i$，得到 k 个 n 位数 $C''_1, C''_2, \cdots, C''_k$，计算 $S_i \oplus C''_i$，$(i=1,2,\cdots,k)$，并把结果发给 B。A 用 C 的密钥解密 $B'_i$，得到 k 个 n 位数 $B''_1, B''_2, \cdots, B''_k$，计算 $S_i \oplus B''_i$，$(i=1,2,\cdots,k)$，并把结果发给 C。

(6) B 通过将 $C_b$ 与从 A 处接收到的第 b 个数异或，计算出 $S_b$；C 通过将 $B_c$ 与从 A 处接收到的第 c 个数异或，计算出 $S_c$。

举例如下：

A 有 8 个秘密要出售：$S_1=1990$，$S_2=471$，$S_3=3860$，$S_4=1487$，$S_5=2235$，$S_6=3751$，$S_7=2546$，$S_8=4043$，B 想买 $S_7$，C 想买 $S_2$。

(1) A 使用 RSA 算法，她与 B 一起使用的密钥对是：$n=7387$，$e=5145$，$d=777$；与 C 一起使用的密钥对是：$n=2747$，$e=1421$，$d=2261$，并把公开密钥分别告诉 B 和 C。

(2) B 产生 8 个 12 位的随机数：$B_1=743$，$B_2=1988$，$B_3=4001$，$B_4=2942$，$B_5=3421$，$B_6=2210$，$B_7=2306$，$B_8=222$；并把它们告诉 C。

$C_1=1708$，$C_2=711$，$C_3=1969$，$C_4=3112$，$C_5=4014$，$C_6=2308$，$C_7=2212$，$C_8=222$；并把它们告诉 B。

(3) B 想买 $S_7$，所以用 A 给他的公开密钥加密 $C_7$。

$$C_7^e \bmod n = 2212^{5145} \bmod 7387 = 5928$$
$$C_7 = 2212 = 0100010100100$$
$$C_7^e \bmod n = 5928 = 1011100101000$$
$$\mathrm{FBI}(C7, C_7^e \bmod n) = \{0,1,4,5,6\}$$

B 把这个 FBI 发给 C。

C 想买 $S_2$，所以用 A 给他的公开密钥加密 $B_2$，计算 $B_2$ 与加密结果的 FBI，{0,1,2,6,9,10} 发给 B。

(4) B 取 $B_1, B_2, \cdots, B_K$，对索引不在集合{0,1,2,6,9,10}中的每一个位用其代替，例如

$$B_2 = 1988 = \underline{1}11\ \underline{11}1\ \underline{000}100$$
$$B'_2 = 1660 = 011001111100$$

B 将 $B'_1, B'_2, \cdots, B'_k$ 发给 A。

C 取 $C_1, C_2, \cdots, C_8$，对索引不在集合{0,1,4,5,6}中的每一个位用其代替，例如

$$C_7 = 2212 = \underline{010001}010\ \underline{01}00$$
$$C'_7 = 5928 = 1011100101000$$

C 将 $C'_1, C'_2, \cdots, C'_8$ 发给 A。

(5) A 用 B 的秘密密钥来解所有的 $C_i$，并把该结果与 $S_i$ 相异或，例如，$i=7$ 时：

$$5928^{777} \bmod 3787 = 22122546 \oplus 2212 = 342$$

该结果发给 B。

A 用 C 的秘密密钥来解所有的 $B_i$，并把该结果与 Si 相异或，例如，$i=2$ 时：

$$1660^{2261} \bmod 2747 = 1988417 \oplus 1988 = 1555$$

该结果发给 C。

(6) B 用 $C_7$ 与从 A 接收到的第 7 个数相异或，得到 $S_7$：

$$2212 \oplus 342 = 2546$$

C 用 $B_7$ 与从 A 接收到的第 2 个数相异或，得到 $S_2$：

$$1988 \oplus 1555 = 471$$

该协议对任意数目的买主都可以使用。如果 B、C、D 3 人想买秘密，则给每一个买主两个公开密钥，分别用于另外两个人，每一个买主可以从另外的买主那儿得到一组数，然后与 A 一起对每一组数完成该协议，并将其与来自 A 的最终结果相异或，就得到他们要的秘密。

但是，两个不诚实的参与者可以合谋进行欺骗。例如 A 和 C 合谋很容易知道 B 想要的是什么秘密。

该协议如图 5-17 所示。

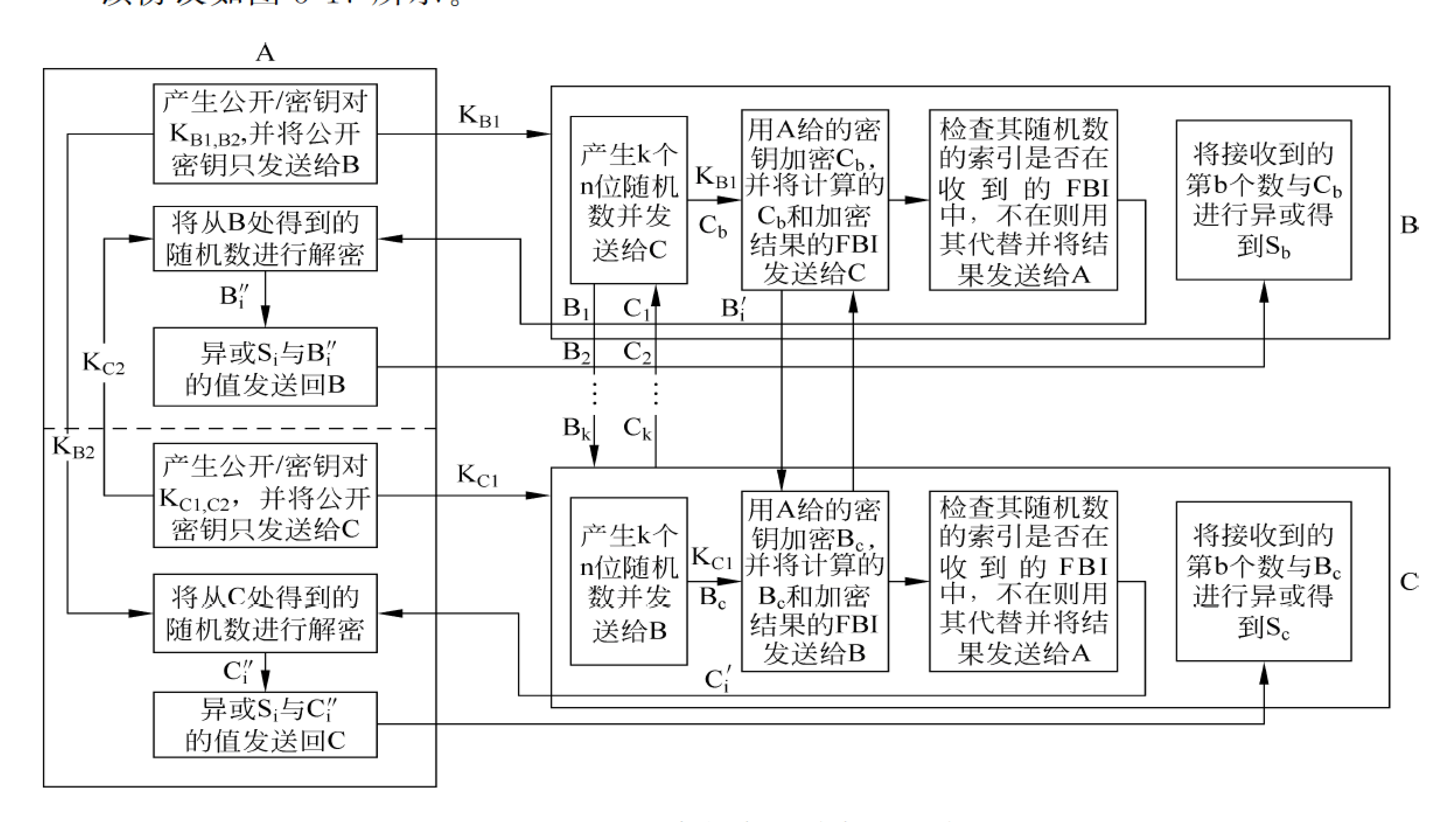



图 5-17　秘密的全泄露或无泄露

## 5.3.13 密钥托管

有时,可能需要将密钥(或对密钥的间接访问)交给那些不直接参与安全通信的人。

下面这段话摘自 Silvio Micali 的专题介绍:

当前,法院授权许可的搭线窃听是防止犯罪并将罪犯绳之以法的有效方法。更重要的是,照我们的观点,通过阻止对正常网络通信的非法使用也防止了犯罪的进一步扩散。因此,法律上比较关心的是,公开密码学的广泛应用可能对犯罪和恐怖组织有很大帮助。实际上,很多议案提议:一个适当的政府机关,在法律允许的情况下,应当可以获得任何通过公共网络进行通信的明文。目前,这个要求可能意味着强迫市民:

(1) 使用弱的密码系统,即有关当局(当然也可以是任何其他的人)经过一定的努力可以破开的密码体制。

(2) 事先把他们的密钥交给当局。这种替代方法会从法律上提醒许多有关的市民,让他们觉得国家安全和法律强制应在隐私之上的话,这并不令人惊奇。

密钥托管(keye scrow)是美国政府的 Clipper 计划和它的托管加密标准的核心。这里,面临的挑战是开发一个密码系统,要保护个人隐私但同时又要允许法院授权的搭线窃听。

托管加密标准通过防篡改的硬件来实现安全性。每个加密芯片有一个唯一的 ID 号和密钥,密钥被分为两部分,并与 ID 号一起由两个不同的托管机构存储。芯片每次加密数据文件,它首先用唯一的密钥加密会话密钥,然后通过通信信道发送加密的会话密钥和它的 ID 号。当一些法律执行机构想用这些芯片中的一个解密加密的信息流时,它监听 ID 号,从托管机关收集适当的密钥,把它们异或起来,解密会话密钥,然后使用会话密钥解密信息流。面对欺诈者,为了使这个方案可行,它可能更复杂。同样的事情能够用软件实现,也可用公开密钥密码实现。

Micali 称自己的思想为公平密码系统(据传美国政府在其托管加密标准中为了使用他的专利花了 1 百万美元。然后,Banker's Trust 购买了 Micali 的专利)。在这些密码系统中,私钥分成许多部分,发给不同机构。类似秘密共享方案,这些机构可集中到一起并重新构造私钥。但是,这些密钥碎片具有一种附加的性质:无需重新构造私钥,便能分别验证这些密钥碎片是否正确。

A 可以产生自己的私钥并给几个托管人每人一部分密钥。这些托管人中没有人能恢复出 A 的私钥。然而,所有这些托管人都能验证他们的那一部分是私钥的有效部分,A 不可能送给一个托管人随机比特串,并希望他带着逃跑了。如果法院授权搭线窃听,有关法律执行机构可以遵照法庭的命令让 n 个托管人交出他们的那一部分密钥。用所有这 n 部分,执行机构重新构造出私钥,并能够对 A 的通信线路进行搭线窃听。另一方面,M 为了能重新构造出 A 的密钥并侵犯她的隐私,将不得不破坏所有 n 个托管人。

协议执行的过程如下:

(1) A 产生出自己的私钥/公钥密钥对,把私钥分成几个公开和秘密部分。

(2) A 送给每个托管人一个公开的部分及对应的秘密部分。这些消息必须加密。她也把公开密钥送给 KDC(密钥分配中心)。

(3) 每个托管人独立完成计算以确认所得到的公开部分和秘密部分都是正确的。每个托管人将秘密部分存放在安全的地方并把公开部分发送给 KDC。

(4) KDC 对公开部分和公开密钥执行另一种计算。假设每一件事都是正确的,KDC 在公开密钥上签名,然后把它送回给 A 或把它邮寄给某处的数据库。

该协议执行过程如图 5-18 所示。

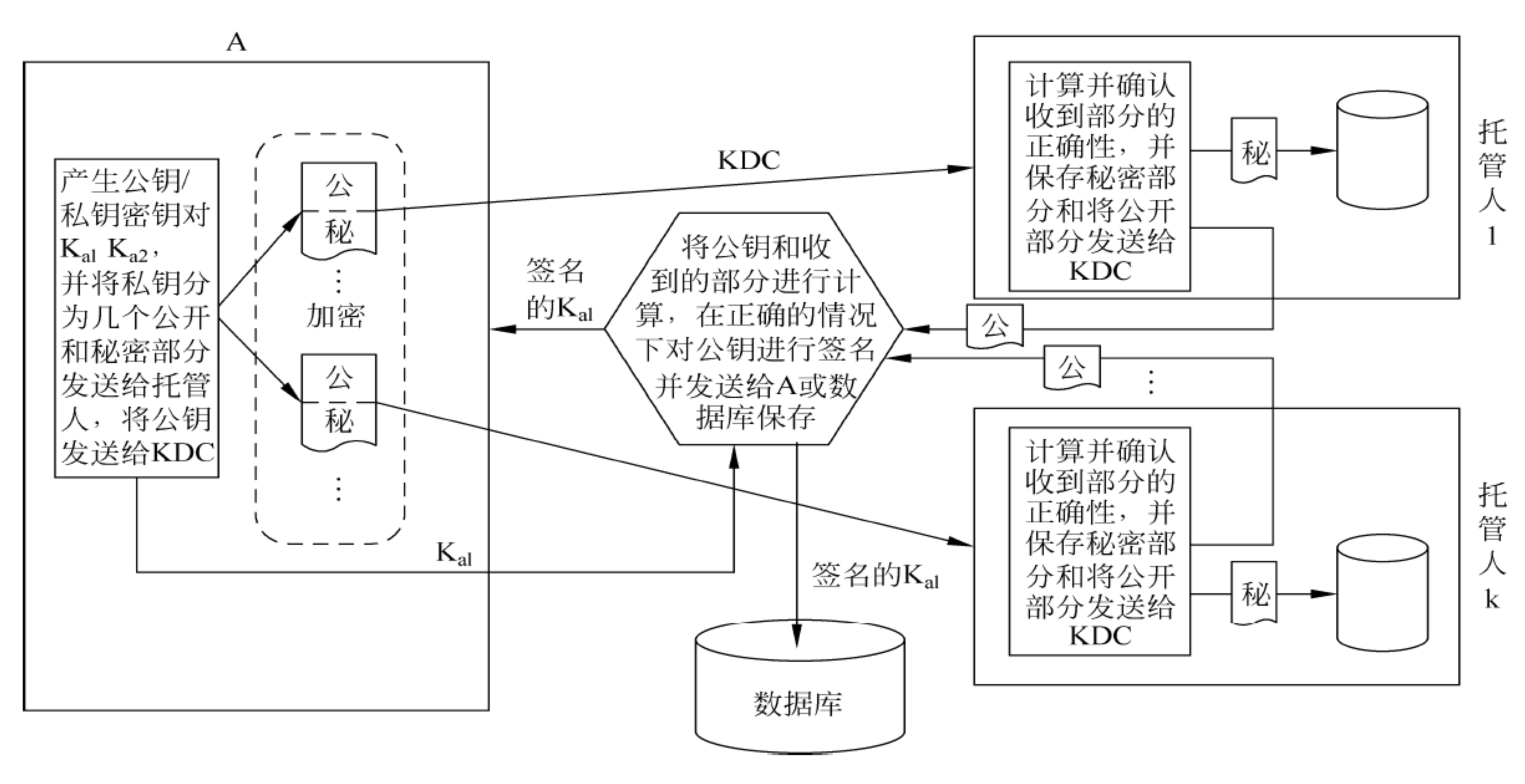



图 5-18 密钥托管

如果法庭要求进行搭线窃听,那么每个托管人就把他或她的那部分交给 KDC,KDC 能重新构造出私钥。在交出密钥前,无论是 KDC 还是任何一个托管人都不能重新构造出私钥,所有托管人一起才能重新构造出这个密钥。

用这种方式能把任何公开密钥密码算法都做成是公正的。Micali 讨论了把门限方案与这个协议结合起来的办法,使得只需要托管人的一个子集(例如,5 个中的 3 个)便能重新构造出私钥。他又讲述了怎样将不经意传输与这个协议结合起来,使得托管人不知道是谁的私钥正在被重新构造。

公平密码系统不是完美的,罪犯可以利用这个系统,使用隐蔽信道把另一个密钥嵌入到自己的那部分中。采用这种方法,使用隐蔽密钥,不用担心法院授权的搭线窃听,他就能够安全地与其他人通信。

### 5.3.13.1 密钥托管的政治问题

除了政府的密钥托管计划外,几个商业密钥托管正在付诸实施。这样就出现了一些问题:对用户来说,密钥托管的好处是什么?

实际上没有任何好处,用户不能从密钥托管得到任何东西。如果他愿意,他可以备份他的密钥。密钥托管保证:即使使用加密,警察也能够窃听他的谈话或阅读他的数据文件;即使使用加密,NSA 不经批准也能够窃听他的国际电话。也许,他将被允许在现在

反对密钥托管的国家使用密码,这似乎是唯一的好处。

密钥托管有相当大的缺陷。用户不得不相信托管机构的安全性程序,以及参与人的诚实。他不得不相信托管机构没有改变他们的策略,政府没有改变他的法律,那些得到密钥的执法机构和托管机构会合法地和负责地做事。设想一个大恐怖分子袭击纽约时,对警察来说,还有什么样的限制不能抛到一边呢?

难于想像托管加密方案工作会像他们的发起人设想的那样没有一些法律的压力。很明显的,下一步是禁止使用非托管加密,这可能是使商业系统付费的唯一办法,并且它肯定是使技术上富有经验的罪犯和恐怖分子使用它的唯一方法。不清楚要使非托管密码成为非法将会遇到什么阻力,或它怎么影响作为研究学科的密码学。

### 5.3.13.2　密钥托管的法律问题

如果有加密数据被破开,托管密钥如何影响用户的责任?如果美国政府试图保护托管结构,是不是有隐含的假设,在用户或托管机构都会危及到秘密的安全时,泄密的一定是用户吗?

对于政府或商业性的密钥托管服务而言,它的整个托管密钥数据库被偷盗了会怎么样?如果美国政府试图对它保持一段时间的沉默又会怎么样呢?很清楚,这会对使用密钥托管的用户愿望产生影响,如果不是自愿的,这样的一些丑闻又将增加政治压力,迫使政府要么让其成为自愿,要么增加复杂的新规定。

更为危险的是现政府的政治对手、对某些情报或警察机构坦率直言的批评家已经被监视多年的丑闻会公诸于世。这可能引起公众强烈反对托管加密的情绪。

如果签名密钥和加密密钥一样被托管,存在更多问题。当局使用签名密钥执行操作反对可疑罪犯能否被接受?基于托管密钥签名的真实性在法庭上是否会被接受?

### 5.3.13.3　密钥托管的其他问题

密码的全球化导致了另外一些问题,密钥托管的政策在其他国家将会一致吗?跨国公司为了保持与各种地方法律一致,他们必须在每个国家保持单独的托管密钥吗?如果没有某种一致性,密钥托管方案的好处之一(强加密的国际化使用)必将崩溃。

如果有些国家根本不接受托管机构的安全性会怎么样呢?用户在那里怎么做生意呢?他们的数字化合同能得到当地法院的支持吗?或者他们的签名密钥托管在美国的事实会允许他们在瑞士声称别人也可能签署他的电子合同吗?在这些国家做生意的人是否有特殊的弃权呢?

工业间谍又会怎么样呢?没有理由相信那些目前正在为其重要的或政府性质的公司从事间谍活动的国家会放弃在密钥托管加密系统上做手脚。的确,由于没有哪个国家会允许其他国家监视自己的情报工作,所以,托管加密的广泛使用必将可能增加搭线窃听的盛行。

即使具有良好公民权记录的国家,其使用密钥托管只是为了合法追踪罪犯和恐怖分子,但它肯定也用于别的地方以跟踪异己分子、有敲诈勒索倾向的政敌等。数字通信在监视公民的行动、意见、购买和集会等一整套工作上提供的机会比模拟世界可能提供的机会

大得多。

## 5.4 本章重点和难点

本章重点是安全协议的设计和分析方法，设计抗攻击的协议所使用的技术手段。

本科讲授时，建议课堂讲授对协议的典型攻击（5.1.1 节）、对协议的安全性分析（5.1.2 节）、安全协议的缺陷（5.1.3 节）、安全协议的设计原则（5.1.5 节）、中间人攻击（5.2.1 节）、阻止中间人攻击的连锁协议（5.2.2 节）；研究生教学适当增加安全协议的形式化分析（5.1.4 节）。其余各节可以自学。

本章的难点是，理清安全协议的实际例子和设计方法与分析验证之间的技术思路。

## 习题与思考题

1. 对协议的典型攻击有哪几种？试分别举例说明。
2. 协议存在哪些安全缺陷？
3. 阐述协议形式化分析的基本途径。
4. 安全协议的设计有哪些原则？
5. 除了中间人攻击外，常见的还有哪些攻击类型是针对密钥交换协议的？
6. 时间戳用于何种情况？
7. 链接协议用来解决什么问题，存在什么问题？试设计一种协议解决链接协议的问题。
8. 试描述隐蔽信道的通信过程，存在的问题，并设计解决方案。
9. 指出现实中可能的隐蔽信道。
10. 试设计一个隐蔽信道传输协议，并分析它的安全性。
11. 设计并分析一个抗抵赖的数字签名。
12. 举例说明代理签名。
13. 试述密钥托管所带来的一些社会问题。

# 第 6 章 实际使用的安全协议

本章介绍在网络应用中正在实际使用的一些安全协议。

本章共有七个小节：第 6.1 节首先介绍现实网络协议需要考虑的因素；第 6.2 节介绍一次性登录技术；第 6.3 节介绍电子支付协议；第 6.4 节介绍公钥基础设施；第 6.5 节介绍防火墙技术；第 6.6 节介绍 VPN 技术；第 6.7 节是本章重点和难点分析。

## 6.1 现实协议需要考虑的因素

前面各章的协议，描述的是一个理想的状态，而在实际应用中，一些理论上不错的方案却在实际中无法实现，主要有 3 个方面的原因，即：计算环境相关的问题，组织结构的问题和电子身份认证方法的问题。

### 6.1.1 与计算环境相关的问题

当前，计算机环境的主要问题是很少有系统在进行安全设计的时候参考了那些普遍通用的认证方法。所以，当新的系统实现了自己的认证和访问控制后，与旧的认证和访问控制机制基本上就无什么互操作性可言。

另一方面，在所有安全解决方案中，“信任”是主要元素。但是，当前的计算机系统不能被信任。它们要么存在严重安全漏洞和错误，要么不能经受恶意攻击。在这些不可靠的部件上运行安全软件，构筑安全平台，是一个挑战。

第三个问题是系统管理员往往对复杂的网络环境中所有服务和配置缺乏足够认识。

### 6.1.2 与组织结构相关的问题

访问授权的规则需要规定哪些资源是个体用户可以或不能访问的。为了简化，用户中类似的需要和权利划分成组。管理不同组的用户是件费力的事情，如果用户转移到别的部门，那么他的访问控制权限也应得到及时反映。尤其在一些基于小组进行活动的组织中，工作上的频繁变动时有发生。但是，部门中组与组间的界定，往往是模糊的。

而当有组织结构上的模糊与计算机环境的繁复相结合时，结果是显然的。系统安全主管必须应付一个异常复杂的情况。

### 6.1.3 与电子身份相关的问题

登录到一个系统的基础是电子身份认证。基本上每种解决方案都有一些利弊。

传统方式也是运用最为广泛的基于口令的认证，即 password authentication，这种方

式的弱点是被猜测和监听。甚至有很多口令被记在笔记本上或就在计算机附近。

对于口令认证的改进是一次性口令。顾名思义,仅使用一次性的口令,可以极大降低监听带来的危险。

电子身份也可以基于智能卡或加密算法,例如 RSA。卡和私钥将被口令加密保护。

一旦实施了安全认证,下一个挑战就是使每个系统接受相同的电子身份,以便为用户产生凭证,并且自动把它们传递给所有需要的服务。这可能是需要实现的最艰巨的部分。

## 6.2 一次性登录技术

一次性登录技术(singlesign-on,SSO),就是通过用户的一次性认证登录,获得需访问系统和应用软件的授权。在此条件下,管理员无需修改或干涉用户登录就能方便地实施希望得到的安全控制。这是一个为了能够在分布式计算机环境中,安全和方便的认证用户而产生的课题。

随着信息技术和网络技术的发展,各种应用服务的不断普及,用户每天需要登录到许多不同的信息系统,如网络、邮件、数据库、各种应用服务器等。每个系统都要求用户遵循一定安全策略,比如要求输入用户 ID(标识符)和口令。随着用户需要登录系统的增多,出错的可能性就会增加,受到非法截获和破坏的可能性也会增大,安全性就会相应降低。而如果用户忘记了口令,不能执行任务,就需要请求管理员帮助,并只能在重新获得口令之前等待,造成了系统和安全管理资源的开销,降低了工作效率。为避免这种尴尬,牢记登录信息,用户一般会简化密码,或者在多个系统中使用相同口令,或者创建一个口令"列表",这些都是会危及信息保密性的几种习惯性做法。

当这些安全风险逐步反映出来,管理员增加一些新安全措施的时候,这些措施却在减少系统的可用性,并且会增大系统管理的复杂度。

因此,在市场上提出了这样的需求:网络用户可以基于最初访问网络时的一次身份验证,对所有被授权的网络资源进行无缝的访问。从而提高网络用户的工作效率,降低网络操作的费用,并提高网络的安全性。

一次性登录技术有多种解决方案,可分为通用标准解决方案和实用化的解决方案两类。前者包括通用安全服务应用程序接口、开放软件基金会的分布式计算环境、嵌入式认证模块;后者包括基于经纪人的解决方案(包括 Kerberos 协议、欧洲安全多环境应用系统、IBM Krypto Knight 等)、基于代理的解决方案、基于口令的解决方案、代理和经纪人相结合的解决方案,以及基于网关的解决方案等。

这里只介绍通用标准解决方案。

### 6.2.1 通用安全服务应用程序接口

关于认证和密钥分配系统经常遇到的一个问题是,由于它要求对应用系统本身作出改动,所以经常受到冷遇。考虑到这一点,对一个认证和密钥分配系统来说,提供一个标准化的安全应用程序接口就显得格外重要。能做到这一点,开发人员就不必再为增加很少的安全功能而对整个应用程序动大手术了。因此,认证系统设计领域内最主要的进展

之一就是制定了标准化的安全应用程序接口，即通用安全服务应用程序接口（generic security service application program interface，GSS-API）。德州 Austin 大学的研究者开发的安全网络编程（SNP），对 GSS-API 接口进行了进一步的封装，使与网络安全性相关的编程更加方便。

一个典型的 GSS-API 调用者是通信协议本身，调用 GSS-API，用可信性、完整性和机密性的安全服务来保护他的通信（如图 6-1 所示），例如 Kerberos。这就是 GSS-API 可以在不同安全服务和应用程序被使用的原因，包括 SSO。GSS-API 的目的是提供隐蔽、特定的内在安全机制的一个接口。这可以帮助不同应用程序之间有更好的互操作性。

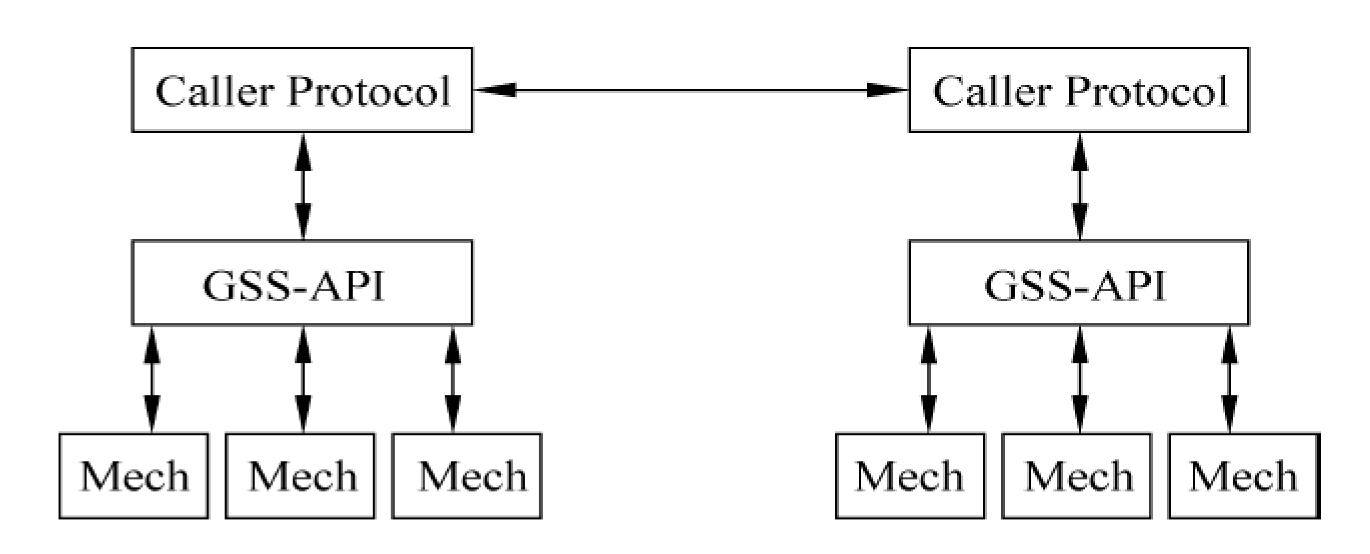



图 6-1　GSS-API

GSS-API 的设计假定并强调以下几个基本目标：

（1）**机制独立**：GSS-API 定义了一个接口来使用密码技术实现强壮的认证和其他安全服务——在独立于特定的底层机制的通用层上。例如，GSS-API 提供的服务可以用密钥技术实现（例如 Kerberos）或者使用公钥技术实现（例如 X.509）。

（2）**协议环境独立**：GSS-API 独立于使用它的通信协议组，允许在多种协议环境中使用。在进行调用的协议和 GSS-API 的应用中，加入一个面向特定通信协议（如 RPC）的中介，可以保持 GSS-API 功能的启用和协议通信的启用之间同步。

（3）**协议联合的独立**：GSS-API 安全上下文构造是独立于通信协议相关的构造的。这个特点允许单独的 GSS-API 实现可以被多种协议模块使用，以利于调用这些模块的应用程序。同时，GSS-API 服务也可以被应用程序直接调用，完全独立于协议关联。

（4）**适应多种实现**：GSS-API 客户不是被限制存在于实现 GSS-API 的系统定义的 TCB（trusted computing base）范围内；安全服务以一种既适应 Intra-TCB 调用，又适用 Extra-TCB 调用的方式加以说明。

关于 GSS-API 的更详细定义，可以参看 RFC2025 和 RFC2078。

## 6.2.2　开放软件基金会分布式计算环境

开放软件基金会（OSF）的分布式计算环境（DCE）是一个被广泛接受的解决方案，用于开发和部署安全的、企业级的分布式计算应用，提供网络安全、透明的服务分配和跨平台通信的能力，允许在一个异构环境中快速设计基于“主/从”或“对等”结构的应用。它能方便地对网络提供最佳性能和可靠保护。

因为 DCE 是由主流操作系统厂商的行业协会所支持的，所以这个标准在很多计算平台上都得到了广泛支持。DCE 核心功能现在已经被几乎所有 Unix 系统及其变种所

支持,并且,在PC操作系统日益普及的今天,DCE核心服务也在PC机上变得越来越普遍。

DCE的认证管理服务是集成了基于DES私人密钥加密技术和MIT开发的Kerberos技术的身份验证。这是一种企业级的安全解决方案,它使企业能为网络资源的使用提供安全。保护管理和通过企业Intranet的用户和通过Internet的远程用户都可以有控制地访问这些资源。

DCE对于安全涉及到4个方面:

(1) 认证(authentication)。

(2) 安全通信(secure communications)。

(3) 授权(authorization)。

(4) 审计(auditing)。

### 6.2.3 嵌入式认证模块

嵌入式认证模块(pluggable authentication modules,PAM)是由Sun公司提出的一种用于实现应用程序的认证机制。其核心是一套共享库,目的是提供一个框架和一套编程接口,将认证工作由程序员交给管理员。

PAM允许管理员在多种认证方法之间作出选择,它能够改变本地认证方法而不需要重新编译与认证相关的应用程序,同时也便于向系统中添加新的认证手段。

PAM最初是集成在Solaris中,目前已移植到其他操作系统中,如Linux、Sun OS、HP-UX9.0等,并在Linux中得到广泛的应用。

PAM整个框架结构如图6-2所示。

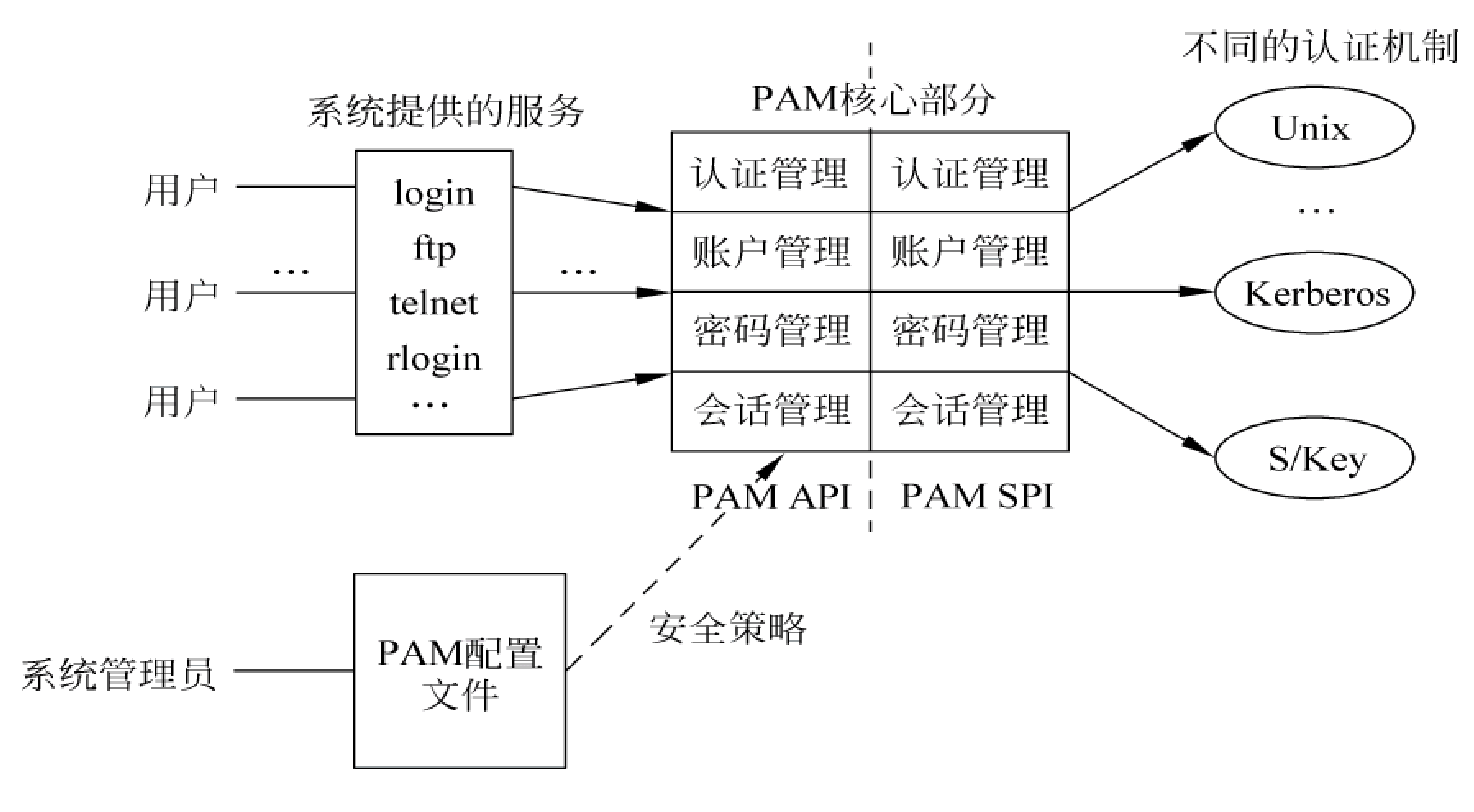



图6-2 PAM框架结构

PAM设计目标是:

(1) 管理员可以选择认证方式,从简单的密码到智能卡系统。

(2) 可以为不同程序配置不同认证机制。如对telnet使用S/Key认证。而本机的login默认使用一般的Unix Password。

（3）支持程序显示方式的需求。如 login 需要基于终端的显示，而 dtlogin 需要 X 显示，而“ftp”和“telnet”需要透过网络来认证。

（4）支持为一个程序配置同时使用多种认证机制。

（5）用户在使用多种认证机制时，同一个密码不必敲入多次。

（6）用户在认证时需要输入多个密码。

（7）当底层的认证机制改变时，上层软件不需要修改。

（8）结构为 system authentication 提供一个 pluggable_model。

（9）必须能满足现有的服务需要。

PAM 功能包括：

（1）加密口令（包括 DES 以外的算法，如 AES 等）。

（2）对用户进行资源限制，防止分布式拒绝服务（DoS）攻击。

（3）允许随意 Shadow 口令。

（4）限制特定用户在指定时间从指定地点登录。

（5）引入概念“client plug-in agents”，使 PAM 支持 C/S 应用中的机器，机器认证成为可能。

PAM 为开发更有效的认证方法提供了便利。在此基础上可以很容易地开发出替代常规的用户名加口令的认证方法，如智能卡、指纹识别等认证方法。

## 6.3 电子支付协议

电子支付是电子商务中的重要环节，涉及到用户与银行等金融部门的交互和接口，其安全性是整个电子商务安全中很重要的一个方面。目前，信用卡是电子商务交易中首选的支付方式，而支持信用卡的安全协议也成为目前电子支付技术的热点。

电子商务安全主要涉及到交易身份的可认证性、信息的保密性、完整性和不可否认性等。

#### 1. 身份的可认证性（authenticity）

指信息的接收者能确认信息的真实来源。在传统交易中，双方可以通过签名、印章等一系列有形的身份凭证来确认身份。然而，在进行网上交易时，交易双方可能素昧平生且相隔千里，如果不采取任何新的保护措施，验证对方的身份，就要比传统商务更容易引起假冒、诈骗等违法活动。

因此，电子交易的首要安全需求就是要保证身份的可认证性。这就意味着，在双方进行交易前，首先要能确认对方身份，要求交易双方的身份不能被假冒或伪装。

#### 2. 信息的保密性（confidentiality）

在传统贸易中，交易双方一般通过面对面或其他可靠通信渠道进行信息交换，达到保守商业机密的目的。而电子商务是建立在一个开放的网络环境下，当交易双方通过 Internet 存储和交换信息时，如果不采取适当保密措施，就有可能造成敏感商业信息泄

露，导致商业上的巨大损失。所以，信息的保密性是电子商务一个重要的安全需求。这就意味着，电子商务的交易方一定要对重要信息进行加密，即使别人截获或窃取了数据，也无法识别信息的真实内容，这样就可以使商业机密难以泄露。

**3．信息的完整性（integrity）**

电子商务简化了贸易过程，减少了人为干预，同时也带来维护贸易各方商业信息的完整、统一的问题。由于数据输入时的意外差错或欺诈行为，可能导致贸易各方信息的差异。此外，数据传输过程中信息丢失、信息重复或信息传送的次序差异也会导致贸易各方信息不同。贸易各方信息的完整性将影响贸易各方的交易和经营策略，保持贸易各方信息的完整性是电子商务应用的基础。因此，要预防对信息的随意生成、修改和删除，同时要防止数据传送过程中信息的丢失和重复并保证信息传送次序一致。

在电子商务中网络还可能面临被动攻击和主动攻击两类威胁。被动攻击就是不修改任何交易信息，但通过截获、窃取、观察、监听等方式获得有价值的情报。而主动攻击就是篡改交易信息，破坏信息的完整性和有效性，以达到非法的目的。信息的保密性是针对被动攻击一类威胁提出的安全需求，但它并不能避免针对网络所采用的主动攻击一类的威胁。因此，保证信息的完整性也是电子商务活动中一个重要的安全需求。这就意味着，交易各方都能验证收到的信息是否完整，即信息是否被篡改过，或在数据传输过程中是否出现信息丢失等差错。

**4．不可否认性（non-reputation）**

不可否认性也称为可追究性。不可否认性是指贸易方对其所作的动作不可抵赖。在传统贸易中，贸易双方通过在交易合同等书面文件上手写签名或印章，预防抵赖行为的发生。但在无纸化的电子交易中，人们不可能再通过传统方法来预防抵赖行为，必须采用新的技术防止电子商务的抵赖行为，否则就会引起商业纠纷，使电子商务无法顺利进行。因而，保证交易过程中的不可否认性也是电子商务安全需求中的一个重要方面。这就意味着，在电子交易通信过程各个环节中都必须是不可否认的，即交易一旦达成，各方不能否认他所发送的或接收的信息。

**5．公平性（fairness）**

公平性是指如果交易成功，交易双方应该获得他们要取得的电子货物。如果交易失败，交易双方谁也不能获得要取得的电子货物。这一点也是进行电子商务活动的客户比较关心的。

**6．匿名性（anonymity）**

一般地，用户在商业交易中都希望保护自己的隐私，不愿泄露自己的身份、购物习惯、购物品种和数量等信息。在现金交易中一般可以很有效地保护顾客的身份。例如，钞票虽有号码，但通常不会暴露使用者是谁。所以匿名性是一种商业交易中的要求。

为了保障电子商务的安全性，除了需要采用防火墙、防病毒和防攻击等网络安全措施

外，一些公司和机构还制定了电子商务的安全协议，来规范在 Internet 上从事商务活动的流程。

典型的电子商务安全协议主要有 SSL（安全套接层）协议和 SET（安全电子交易）协议。

## 6.3.1　安全套接层协议

SSL（secure sockets layer）协议最初是由 Netscape 公司研究制定的安全协议。该协议向基于 TCP/IP 的客户/服务器应用程序提供客户端和服务器的认证、数据完整性及信息机密性等安全措施。该协议通过在应用程序进行数据交换前交换 SSL 初始握手信息来实现有关安全特性的审查。在 SSL 握手信息中采用了 DES、MD5 等加密技术来实现机密性和数据完整性。当然，也可使用 AES、ECC 等高级加密技术实现，并采用 X. 509 的数字证书实现认证。

SSL 协议位于 TCP/IP 协议与各种应用层协议之间，通过它发信方和收信方在建立 TCP 连接之后，可以协商建立安全通信环境，具体包括身份认证（通过对对方证书的验证来实现）、密码算法和密钥的协商、MAC 秘密的协商等。安全环境建立起来之后，就可以收发 SSL 数据（SSL 记录），如图 6-3 所示。

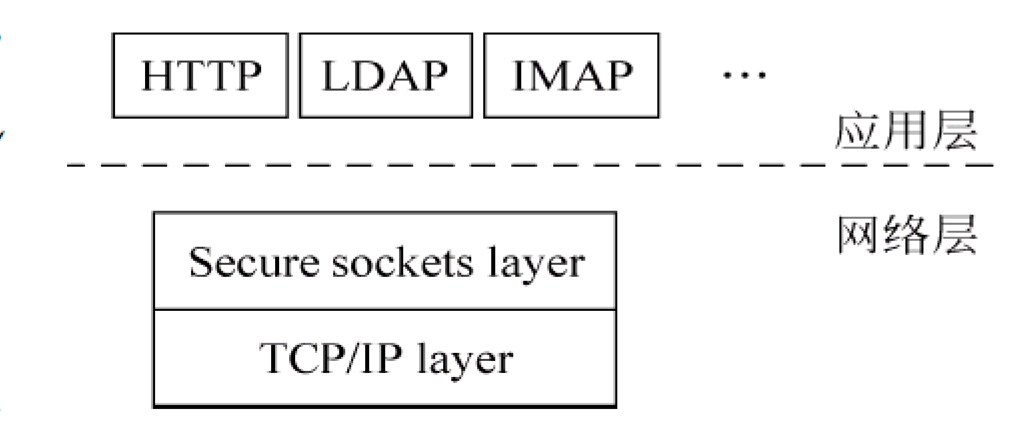
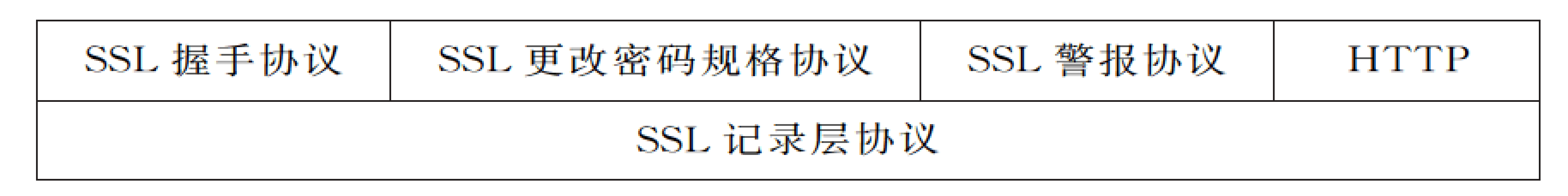



图 6-3　SSL 协议的位置

SSL 不是一个单独的协议，而是两层协议，即握手层和记录层。SSL 握手协议描述建立安全连接的过程，在客户和服务器传送应用层数据之前，完成诸如加密算法和会话密钥的确定、通信双方的身份验证等功能。SSL 记录协议则定义了数据传送的格式。这样，应用层通过 SSL 协议把数据传给传输层时，已是加密后的数据，此时 TCP/IP 协议只需负责将其可靠地传送到目的地，弥补了 TCP/IP 协议安全性较差的弱点，如图 6-4 所示。



| SSL 握手协议 | SSL 更改密码规格协议 | SSL 警报协议 | HTTP |
|---|---|---|---|
| SSL 记录层协议 | | | |

图 6-4　SSL 协议

（1）SSL 记录协议（SSL record protocol）：建立在可靠的传输协议（如 TCP）之上，为高层协议提供数据封装、压缩、加密等基本功能的支持。

（2）SSL 握手协议（SSL handshake protocol）：建立在 SSL 记录协议之上，用于在实际数据传输开始前，通信双方进行身份认证、协商加密算法、交换加密密钥等。

SSL 安全协议主要提供 3 个方面的服务：

（1）认证用户和服务器，使得它们能够确信数据将发送到正确的客户机和服务器上。

（2）加密数据以隐藏传送的数据。

（3）维护数据的完整性，确保数据在传输过程中不被改变。

SSL 的运行步骤包括 6 步：

（1）接通阶段：客户通过网络呼叫服务商，服务商回应。

（2）密码算法交换阶段：客户与服务商之间交换双方认可的密码算法。一般选用 RSA 密码算法，也有的选用 Diffie-Hellman 和 Fortezza-KEA 密码算法。

（3）会谈密钥阶段：客户与服务商之间产生彼此交谈的会谈密钥。

（4）检验阶段：检验服务商取得的密码。

（5）客户认证阶段：验证客户的可信度。

（6）结束阶段：客户与服务商之间相互交换结束信息。

当上述步骤完成之后，两者间的资料传送就会加密，到另外一端收到资料后，再将编码后的资料还原。即使盗窃者在网络上取得编码后的资料，如果没有原先编制的密码算法，也不能获得可读的有用资料。

在电子商务交易过程中，由于有银行参与，按照 SSL 协议，客户购买的信息首先发往商家，商家再将信息转发银行，银行验证客户信息的合法性后，通知商家付款成功，商家再通知客户购买成功，并将商品寄送给客户。

SSL 安全协议是国际上最早应用于电子商务的一种网络安全协议，至今仍然有许多网上商店在使用。在点对点的网上银行业务中也经常使用。该协议已成为事实上的工业标准，并被广泛应用于 Internet 和 Intranet 的服务器产品和客户端产品中。如 Netscape 公司、Microsoft 公司、IBM 公司等 Internet/Intranet 网络产品的公司已在使用该协议。

SSL 已广泛应用于电子商务领域，主要是因为它能为两个通信实体提供认证、数据的保密性和完整性服务。

（1）用户和服务器的合法性验证。利用证书技术和可信的第三方 CA，可以让客户机和服务器相互识别对方的身份。SSL 要求证书持有者在握手时相互交换数字证书，通过验证来保证对方身份的合法性。

（2）数据的保密性。SSL 所采用的加密技术既有对称密钥技术，也有公开密钥技术。SSL 客户机和服务器通过密码算法和密钥的协商建立起一个安全通道，在此安全通道中传输的所有信息将经过加密处理，这样，网络中的非法窃听者所获取的信息都将是无法识别的密文信息。

（3）数据的完整性。SSL 利用密码算法和 Hash 算法，通过对传输信息特征值的提取来保证信息的完整性，确保要传输的信息全部完整、准确无误地到达目的地。

### 6.3.2 安全电子交易协议

安全电子交易（secure electronic transactions，SET）标准是为了在 Internet 上进行在线交易时保证信用卡支付的安全而设立的一个开放的规范。它是由 Visa International、Master Card International 两大信用卡公司与 IBM、Microsoft、Netscape、GTE、VeriSign、SAIC、Terisa 等厂商合作开发的。

SET 是一种基于消息流的协议。它采用公开密钥密码体制和 X.509 数字证书标准，主要应用于电子商务的 B to C 模式。与 SSL 相比，SET 不仅加密两个端点间的单个会

话，还可以加密和认定三方之间的多个信息。

在 SET 协议中主要定义以下内容：

(1) 加密算法的应用。

(2) 证书消息和对象格式。

(3) 购买消息和对象格式。

(4) 请款消息和对象格式。

(5) 参与者之间的消息协议。

SET 协议主要使用的技术包括：对称密钥加密、公开密钥加密、Hash 算法、数字签名、数字信封以及数字证书等技术。它提供消费者、商家和银行之间的认证，确保交易数据的安全性、完整可靠性和交易的不可否认性，特别是保证不将消费者银行卡号暴露给商家等优点，使其成为目前公认的信用卡/借记卡的网上交易的国际安全标准。

SET 主要由 3 个部分组成，分别是 SET 业务描述、SET 程序员指南和 SET 协议描述。SET1.0 版已经公布并可应用于任何银行支付服务。

SET 采用 RSA 公开密钥体系对通信双方进行认证。利用 AES、RC4 或任何标准对称加密方法进行信息的加密传输，并用 Hash 算法来鉴别消息真伪，有无篡改。在 SET 体系中有一个关键的认证机构 CA(certificate authority)，CA 根据 X.509 标准发布和管理证书。

### 1. SET 安全协议运行的目标

SET 是一个基于可信的第三方认证中心的方案，它要实现的主要目标是：

(1) 保障付款安全：确保付款资料之隐密性及完整性，提供持卡人、特约商店、收单银行之认证，并定义安全服务所需之演算法及相关协定。

(2) 确定应用之互通性：提供一个开放式的标准，明确定义细节，以确保不同厂商开发之应用程序可共同运作，促成软件互通；并在现存各种标准下构建该协定，允许在任何软硬件平台上的执行，使标准达到相容性与接受性的目标。

(3) 达到全球市场的接受性：在容易使用与对特约商店、持卡人影响最小的前提下，达到全球普遍性。允许在目前使用者的应用软件下，嵌入付款协定的执行，对收单银行与特约商店、持卡人与发卡银行间的关系，以及信用卡组织的基础构架改变最少。

为此，SET 协议要做的事情有：

(1) 保证信息在因特网上安全传输，防止数据被黑客或被内部人员窃取。

(2) 保证电子商务参与者信息的相互隔离。客户的资料加密或打包后通过商家到达银行，但是商家不能看到客户的账户和密码信息。

(3) 解决多方认证问题，不仅要对消费者的信用卡认证，而且要对在线商店的信誉程度认证，同时还有消费者、在线商店与银行间的认证。

(4) 保证网上交易的实时性，使所有支付过程都是在线的。

(5) 效仿电子数据交换(electronic data interchange，EDI)贸易的形式，规范协议和消息格式，促使不同厂家开发的软件具有兼容性和互操作功能，并且可以运行在不同硬件和操作系统平台上。

### 2. SET 的运作方式

SET 的运作是通过 4 个软件组件完成的，分别在持卡人、商家、支付网关以及认证中心 CA 的计算机中运行，共同完成整个 SET 交易服务。

(1) 电子钱包：安装在持卡人端的软件组件，称为电子钱包。用户在使用电子钱包进行网上购物时，只需确认订单已发送给商家即可，其他功能电子钱包会自动完成。

(2) 支付网关：是商家和银行间信息传输和转换的桥梁，利于安全通信。

(3) 认证中心：在基于 SET 协议的电子商务体系中起着重要作用，负责为参与交易的各方签发证书，进行身份验证。

SET 是一个基于可信的第三方认证中心的方案，它的核心在于数字证书，它提供简单的方法确保进行电子交易的人们能够互相信任。所以，在使用基于 SET 协议的银行卡进行网上支付前，持卡人、商家、支付网关都必须向认证中心申请数字证书，这样才能参与 SET 交易。

SET 最主要的使用对象在消费者与商店之间，商店与收单银行(付款银行)之间。其运作方式简述如下：

(1) 在消费者与特约商店之间，由持卡人在消费前先确认商店的合法性，由商店出示它的证书。

(2) 持卡人确认后即可下订单，其订单经消费者以数字签名(digital signature)的方式确认，而消费者所提供的信用卡资料则另由收单银行以公钥予以加密。这里，特约商店会收到两个加密过的资料，其中一个是订单资料，另一个是关于支付的资料，特约商店可以解密前者，但无法解密后者，避免特约商店搜集或滥用持卡人消费资料。

(3) 特约商店将客户的资料连同自己的 SET 证书发给收单银行，向银行请求交易授权及授权回复。

(4) 收单银行会同时监视两个证书来确定是否为合法的持卡人及特约商店。收单银行会由支付系统网关(payment gateway)来解密，核对资料无误后，再连线到传统的网络(比如 Visa 或 Master Card)做交易授权及清算。

(5) 授权确认后由特约商店向消费者再行确认订单，交易完成。

(6) 至于特约商店与收单银行间，则基于该授权为请款之要求并由银行付款。

图 6-5 是这个过程的图示化表示。

在这个过程中可以看到，CA 扮演了系统中很重要的角色。SET 标准着重点是其交易安全及隐密性。其中，证书(digital certificate)为其核心，它提供简单的方法来确保进行电子交易的人们能够互相信任。信用卡组织提供数字证书给发卡银行，然后发卡行再提供证书给持卡人；同时，信用卡组织也提供数字证书给收单银行，然后收单银行再将证书发给特约商店。在进行交易的时候，持卡人和特约商店两边符合 SET 的规格软件，会在资料交换前分别确认双方身份，也就是检查由授权的第三者所发给的证书。

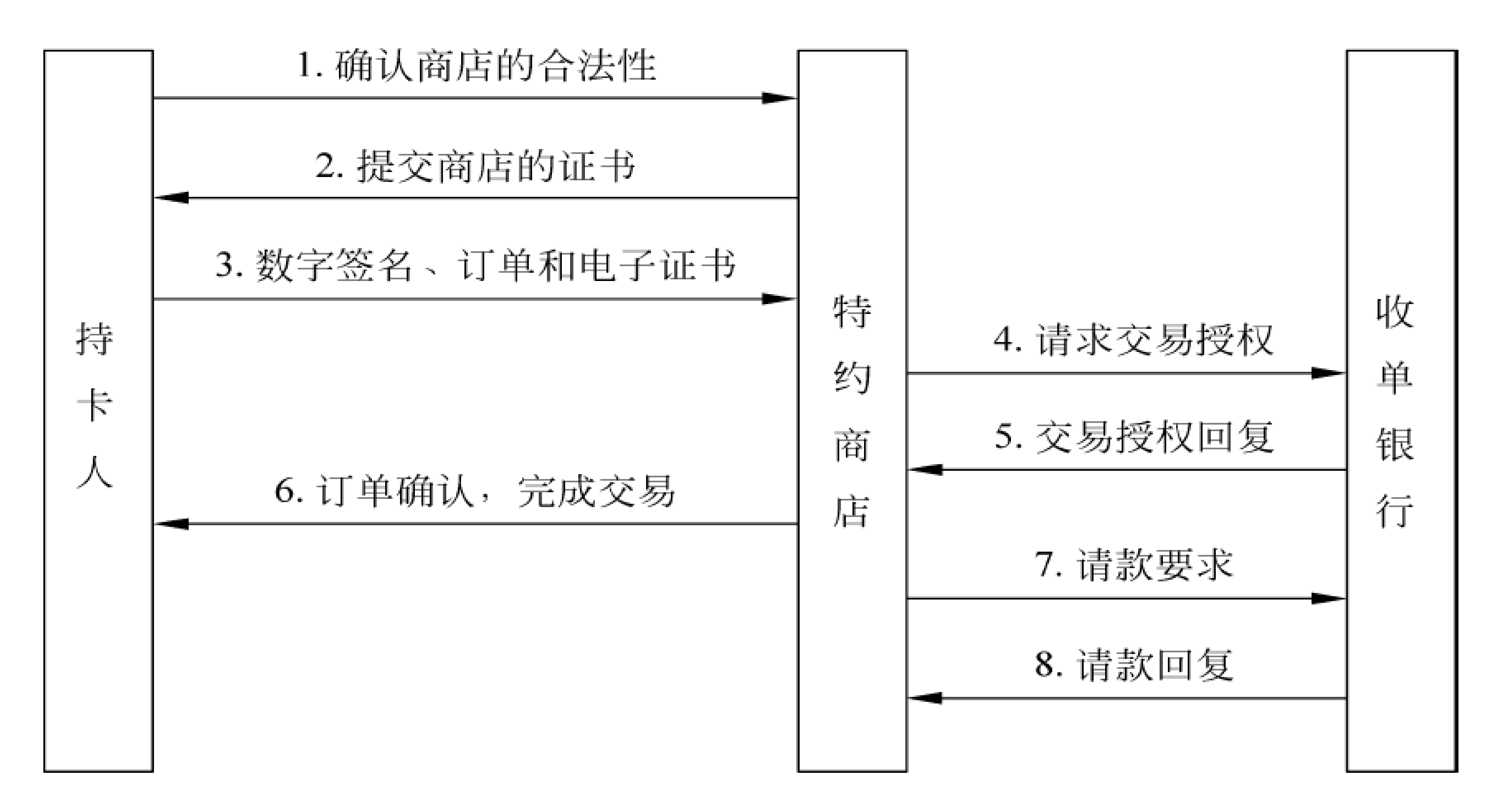



图 6-5　SET 的简易流程示意图

### 3. SET 安全协议涉及的范围

SET 协议规范所涉及的范围有：

(1) 消费者。包括个人消费者和团体消费者，按照在线商店的要求填写订货单，通过由发卡银行发行的信用卡进行付款。

(2) 在线商店。提供商品或服务，具备相应电子货币使用的条件。

(3) 收单银行。通过支付网关处理消费者和在线商店之间的交易付款问题。

(4) 电子货币(如智能卡、电子现金、电子钱包)发行公司以及某些兼有电子货币发行的银行。负责处理智能卡的审核和支付工作。

(5) 认证中心(CA)。负责确认交易双方的身份，对厂商的信誉度和消费者的支付手段进行认证。

### 4. SET 的认证

在用户身份认证方面，SET 引入了证书(certificates)和证书管理机构(certificates authorities)机制。

(1) 证书

证书就是一份文档，它记录了用户的公共密钥和其他身份信息。在 SET 中，最主要的证书是持卡人证书和商家证书。

持卡人证书(cardholder certificates)：它实际上是支付卡的一种电子化表示。它是由金融机构以数字签名形式签发的，不能随意改变。持卡人证书并不包括账号和终止日期信息，取而代之的是用单向哈希算法根据账号、截止日期生成的一个编码，如果知道账号、截止日期、密码值即可导出这个码值，反之不行。

商家证书(merchant certificates)：表示可接受何种卡来进行商业结算。它是由金融机构签发的，不能被第三方改变。在 SET 环境中，一个商家至少应有一对证书。一个商家也可以有多对证书，表示它与多个银行有合作关系，可以接受多种付款方法。

除了持卡人证书和商家证书以外，还有支付网关证书(payment gateway certificates)、银行证书(acquirer certificates)、发卡机构证书(issuer certificates)。

(2) 证书管理机构

CA 是受一个或多个用户信任，提供用户身份验证的第三方机构。证书一般包含拥有者的标识名称和公钥，并且由 CA 进行过数字签名。

CA 的功能主要有：接收注册请求，处理、批准/拒绝请求，颁发证书。用户向 CA 提交自己的公共密钥和代表自己身份的信息(如身份证号码或 E-mail 地址)，CA 验证了用户的有效身份之后，向用户颁发一个经过 CA 私有密钥签名的证书。

(3) 证书的树形验证结构

在两方通信时，通过出示由某个 CA 签发的证书来证明自己的身份，如果对签发证书的 CA 本身不信任，则可验证 CA 的身份，依次类推，一直到公认的权威 CA 处，就可确信证书的有效性。SET 证书正是通过信任层次来逐级验证的。

通过 SET 的认证机制，用户不再需要验证并信任每一个想要交换信息的用户的公共密钥，而只需要验证并信任颁发证书的 CA 的公共密钥就可以了。

#### 5. SET 标准的应用

由于设计合理，SET 协议得到了 IBM、HP、Microsoft、Netscape、VeriFone、GTE、VeriSign 等许多大公司的支持，目前已获得互联网工程任务组(the internet engineering task force，IETF)标准的认可，成为企业对个人(B2C)业务事实上的工业标准。

在国际上，已经有很多银行建立了支付网关，接受 SET 方式支付；在国内，中国银行已经采用 SET 方式，建立了认证中心和支付网关。现在，在北京和上海，用户都可以使用长城电子借记卡进行人民币支付，同时，中国银行还利用香港中银信用卡(国际)有限公司的支付网关接受外币业务。

从 1996 年开始，34 个国家的 150 多家金融机构制定了 SET 试行方案。从新加坡到旧金山的信用卡公司都开始建造 SET 交易网关，软件厂家则着手开发支持 SET 的应用软件。

总之，安全的电子支付是电子商务中最为核心和复杂的环节，所以，电子支付方式的选择一定要根据电子商务交易的实际情况，并结合安全性和效率等方面进行综合考虑。SSL 安全协议和 SET 安全协议的应用极大地推动了电子支付技术的提高，从而也促进了电子商务的发展。

### 6.3.3 ISI 协议

ISI 支付协议是由 Medvinsky 和 Neuman 提出的，它的目的是付款人 A 向收款人 B 付款，A 保持匿名。

(1) $A \to B: K_{ab}$

(2) $B \to A: \{K_b\}_{K_{ab}}$

(3) $A \to B: \{\{coins\}_{K_{cs}^{-1}}, SK_a, K_ses, S_id\}_{K_b}$

(4) $B \to CS: \{\{coins\}_{K_{cs}^{-1}}, SK_b, transaction\}_{K_{cs}}$

(5) CS→B:$\{\{new_coins\}_{K_{cs}^{-1}}\}_{SK_b}$

(6) B→A:$\{\{amount, Tid, date\}_{K_b^{-1}}\}_{SK_a}$

通过协议的步骤(1)和步骤(2),A 取得 B 的公开密钥。步骤(3),A 将电子货币、希望获得的服务的标识符 S_id 和密钥 K_ses 用 B 的公开密钥加密后传送给 B。通过 $K_{cs}$,B 可以验证电子货币的有效性。然后,B 通过步骤(4)将电子货币发送给货币服务方(Currency Server,CS),如果这笔钱尚未使用,CS 通过步骤(5)支付给 B。步骤(6),B 将收据传送给 A。

## 6.3.4 First Virtual 协议

First Virtual 协议是美国 First Virtual 公司开发的电子支付协议,它也是用于网上信用卡的安全电子交易协议。与其他协议不同,它采用了非密码学的方法解决安全性问题。

客户先在 First Virtual 建立一个 ID 号,并将自己的信用卡号注册。当需要支付信息费用时,其支付过程如下:

(1) 客户将在 First Virtual 的 ID 号发给商家。

(2) 商家连接 First Virtual 服务器,验证 ID 号的合法性。如果合法,商家把客户所需的信息直接发送给客户。

(3) First Virtual 服务器以 E-mail 形式向客户发送询问信息,征询客户是否愿意付费。

(4) 客户同样以 E-mail 形式回复“是”或“否”。

(5) 如果客户的回复为“是”,First Virtual 通过用户信用卡的代理获得相应的款项。

(6) 90 天后,First Virtual 服务器将款项转给相应的商家。

First Virtual 协议使用了以下几个基本假设:

(1) 客户不知道所索取的信息是否对自己有价值,必须收到信息后再作出支付判断。

(2) 电子发送信息的费用较低,客户不做支付时,商家未受什么损失。

(3) 多数的客户是诚实的,即客户的信用度可以保障。

由此可见,这些条件都与安全性无关。First Virtual 协议的所有安全保障都基于 E-mail 通信体系的安全性和完整性。支付的 90 天延迟是为了防止欺骗的发生,如有这样的情况,客户的款项将被返还。

## 6.3.5 iKP 协议

美国 IBM 公司开发的 iKP 协议(I Key Protocol,i=1,2,3)是一簇安全电子支付协议,该簇协议与现有的商业模型和支付系统基础设施相匹配。协议中涉及 3 个主体,即用户、商家和网关。所有 iKP 协议都基于公开密钥密码学,但它们随拥有公开和秘密密钥对的主体的数目而变化,分别称为 1KP 协议、2KP 协议和 3KP 协议,其安全性和复杂性递增。

其中,1KP 协议是最简单的协议,它只要求网关拥有一对公开和秘密密钥。用户和商家只需拥有网关认证的公开密钥,或经一个权威机构认证的网关公开密钥(该机构通过签名证书使网关的公开密钥合法化)。这就涉及 CA 基础设施,用户通过他们的信用卡号和相关秘密 PIN 进行认证。支付是通过交换用网关的公开密钥加密的信用卡号和 PIN 以及限定的相关信息(如交易量、ID 号等)认证的。1KP 协议不能对用户和商家发送的消息提供非否认性,这就意味着不容易解决支付订购中产生的争端。

2KP 协议要求网关和商家都拥有公开和秘密密钥对,以及公开密钥证书。协议对来自商家发送的消息能提供非否认认证。该协议能使用户无需和任何在线第三方联系就能通过检测他们的证书,验证他们正在和真实的商家进行交易。与 1KP 协议一样,支付订购是通过用户的信用卡号和 PIN 认证的(在传输之前要求加密)。

3KP 协议要求网关、用户和商家三方都拥有公开和秘密密钥对,并提供多方位的安全服务。它对各方涉及的所有消息都提供非否认认证。支付订购是通过用户的信用卡号和 PIN 以及用户的数字签名认证的。该协议要求公开密钥基础设施(PKI)提供用户的公开密钥证书。

特别值得一提的是,iKP 协议只关心支付而未涉及订货与价格协商,协议假设这些内容已由商家和用户事先确定。该协议也没有明确地提供对订购信息的加密,协议假设这类保护已由现有的机制完成,例如 SHTTP、SSL 等。

## 6.3.6 数字现金相关协议

现金有一个问题,它难于搬运,容易传播病菌,并且容易被人偷走。

支票和信用卡大大减少了社会上实际现金的流通量,但实际上不可能完全取消现金。因为毒品贩子和政治家永远不会赞成取消现金。还有,支票和信用卡具有的审计线索,不可能隐瞒把钱给了谁。

另一方面,支票和信用卡又使别人可以侵犯自己的隐私,其过程是以前想像不到的。虽然没有人同意警察跟随其一生,但是警察可以查看相应的金融交易。他们能看到某人在哪里买汽油,在哪里买食物,和谁通了电话——所有这一切都逃不过他们的计算机终端。人们需要有一种方法来保护他们的匿名权、藉以保护他们的隐私。

数字现金又叫电子现金,它是以电子化数字形式存在的现金货币。电子现金比现有的实际现金(纸币和硬币)有更多的优点,实际现金要承担较大的存储风险、高昂传输费用、较大的安全保卫和防伪的投资。其发行方式包括存储性质的预付卡(即电子钱包)和纯电子系统形式的用户号码数据文件等形式。

数字现金具有如下特点:

(1) 银行和卖方之间应有协议和授权关系。

(2) 买方、卖方和 E-cash 银行都需要使用 e-cash 软件。

(3) 适合于小量的交易(Mini Payment)。

(4) 身份验证由 E-cash 本身完成。

(5) E-cash 银行负责买方和卖方之间资金的转移。

(6) 具有现金特点,可以存、取、转让;而且安全性也比较好。

(7) 主要缺点是，就和普通的货币一样，硬盘出故障的时候，如果没有备份，现金就丢失了。

幸好，有一个复杂协议容许消息可以确认，但不可跟踪。举例来说，说客 A 可以把数字现金转移给参议员 B，并使得新闻记者 E 不知道 A 的身份，然后 B 可以把这笔电子货币存入他的账户，即使是银行也不知道 A 是谁。

但是如果 A 试图以用来贿赂 B 的同一笔数字现金来买可卡因，那么她就会被银行检测出来。如果 B 试图把同一笔数字现金存入两个不同账户，他也会被发现——但 A 仍保持匿名。

为了把这种现金与带审计追踪的数字现金(如信用卡)相区别，有时把它叫做匿名数字现金。这类东西有很大的社会需求。随着网上商业交易的发展，商业上日益需要更多的称为网络隐私和网络匿名的东西。比如，人们有很好的理由不愿意通过互联网传送他们的信用卡号。

另一方面，银行和政府似乎不愿放弃目前银行系统提供的审计追踪控制。但是，最终他们不得不放弃。一些可信赖的机构愿意将数字现金转换为真正的现金，所有银行也都将提供数字现金服务。

数字现金协议非常复杂。我们将一步一步来构建一个。要认识到这只是一个数字现金协议，还有其他的。

#### 6.3.6.1　匿名汇票协议

第一个协议是一个有关匿名汇票的简单化的物理协议：

(1) A 准备了 100 张 1000 美元的匿名汇票。

(2) A 把每张汇票和一张复写纸放进 100 个不同信封内，她把这些全部交给银行。

(3) 银行开启 99 个信封并确认每个都是一张 1000 美元的汇票。

(4) 银行在余下的一个未开启的信封上签名，签名通过复写纸印到汇票上。银行把这个未开启的信封交还 A，并从她的账户上扣除 1000 美元。

(5) A 打开信封并在一个商人处花掉了这张汇票。

(6) 商人检查银行的签名以确认这张汇票是合法的。

(7) 商人拿着这张汇票到银行。

(8) 银行验证它的签名并把 1000 美元划入这个商人的账户。

这个协议可以起作用。银行从未看到自己签的那张汇票，故当这个商人把它带到银行时，银行不知道它是 A 的。因为这个签名的缘故银行还是相信它有效。银行相信未开启的汇单是 1000 美元的，那是因为采用分割选择协议的缘故。关于分割选择协议可参考 4.6 节的详细介绍。银行验证了其他 99 个信封，故 A 仅有 1%的机会欺骗银行。

当然，银行对于欺诈将进行足够严厉的惩罚，以致于欺诈与成功的机会相比是不值的。

#### 6.3.6.2　抗双重花费协议

匿名汇票协议防止了 A 在一张汇票上写入比她宣称的更多的钱，但它没有防止 A 将这张汇票复制并两次花掉它。

这叫做“**双重花费问题**”。

为了解决这个问题，需要一个复杂的协议：

(1) A 准备 100 张每张 1000 美元的匿名汇票。在每一张汇票上包含了一个不同且随机的唯一的长字符串，字符串长到足以使另一个人也用它的机会变得微乎其微。

(2) A 把每张汇票都和一张复写纸一起装入 100 个不同的信封，并把它们全交给银行。

(3) 银行开启 99 个信封并确认每张都是一个 1000 美元的汇票，而且所有随机唯一字符串都是不同的。

(4) 银行在余下的一个未开启的信封上签名，签名通过复写纸印到汇票上。银行把这个未开启的信封交回 A，并从她的账户上扣除 1000 美元。

(5) A 打开信封并在一个商人处花掉这张汇票。

(6) 这个商人检查银行的签名以确认汇票是合法的。

(7) 这个商人拿着这张汇票来到银行。

(8) 银行验证它的签名，并检查它的数据库以确认有相同的唯一字符串的汇票先前没有存过。如果没有存过，银行把 1000 美元划到这个商人的账上。银行在一个数据库中记录这个随机字符串。

(9) 如果它先前存过，银行不接受这张汇票。

现在，如果 A 试图使用这张汇票的影印件，或者如果这个商人试图用这张汇票的影印件存款，银行都会发现。

#### 6.3.6.3 防欺骗协议

前面的协议保护了银行不受欺骗者的欺骗，但它没有识别出这些欺骗者。银行不知道是买这张汇票的人(银行不知道是 A)试图欺骗这个商人或者是这个商人试图欺骗银行。

这个协议纠正如下：

(1) A 准备了 100 张每张 1000 美元的匿名汇票。在每一张汇票上包含了一个不同的唯一的随机字符串，字符串长到足以使另一个人也用它的机会变得微乎其微。

(2) A 把每张汇票都和一张复写纸一起装入 100 个不同的信封，并把它们全交给银行。

(3) 银行开启其中 99 个信封并确认每张都是一个 1000 美元的汇票，而且所有随机字符串都是不同的。

(4) 银行在余下的一个未开启的信封上签名，签名通过复写纸印到汇票上。银行把这个未开启的信封交回 A，并从她的账户上扣除 1000 美元。

(5) A 打开信封并在一个商人处花掉了这张汇票。

(6) 这个商人检查银行的签名以确认汇票是合法的。

(7) 商人要求 A 在汇票上写一个随机识别字符串。

(8) A 同意。

(9) 这个商人拿着这张汇票来到银行。

(10) 银行验证签名并检查它的数据库以确认具有相同唯一字符串的汇票先前没有存过。如果没有,银行把 1000 美元划归商人的账上。银行在一个数据库中记下这个唯一字符串和识别字符串。

(11) 如果这个唯一字符串在数据库中,银行拒收这张汇票。接着,它将汇票上的识别字符串同存在数据库中的识别字符串比较。如果相同,银行知道这个商人影印了这张汇票。如果不同,银行知道买这张汇票的人影印了它。

这个协议假设,一旦 A 在汇票上写上这个识别字符串,那个商人就不能改变它。汇票可能有一系列小方格,这个商人会要求 A 用 X 或 O 填充这些小方格。汇票是用如果要抹去字迹就会抹坏的纸做成。

由于商人和银行之间的交互作用发生在 A 花钱之后,故这个商人可能和一张空头汇票牵连在一起。这个协议的具体实现可以要求 A 在商人与银行交互期间在柜台前等着,很像是今天的信用卡交易操作的方式。

A 可能会试图陷害这个商人。她可以第二次花一张汇票的复制,在步骤(7)中给一个同样的识别字符串。如果这个商人不保存一个已收到的汇票的数据库,他就遭到陷害。下面要讲的抗抵赖协议能消除这个问题。

#### 6.3.6.4　抗抵赖的协议

如果证明是买汇票的人试图欺骗这个商人,银行会希望知道那个人是谁。为了做到这一点,要求在实际模拟中利用进入密码技术。

秘密分割技术可以用来在数字汇票中隐藏 A 的名字。

(1) A 对给定数量的美元准备 n 张匿名汇票。每张汇票都包含了一个不同的随机唯一字符串 X,X 有足够长,足以使得有两个字符串相同的机会微乎其微。

在每一张汇票上也有 n 对认证比特字符串 $I_1, I_2, \cdots, I_n$。这些字符串对中,每一个都是按如下产生的:A 建立一个由她的名字、地址以及任何其他银行希望见到的认证信息组成的字符串。接着,她用秘密分割协议将它分成两部分,关于秘密分割协议在前面 4.6 节中已作了详细介绍。然后,她使用一种比特提交协议提交每一部分。

例如,$I_{37}$ 由两部分组成:$I_{37_L}$ 和 $I_{37_R}$。每一部分都是一个可以要求 A 打开的比特承诺分组,其正确打开与否也可以立即验证。任何对如 $I_{37_L}$ 和 $I_{37_R}$,但不是 $I_{37_L}$ 和 $I_{38_R}$ 都会揭示 A 的身份。

每张汇票看起来像这个样子:

总数

唯一字符串:X

认证字符串:$I_1 = (I_{1L}, I_{1R})$

$$I_2 = (I_{2L}, I_{2R})$$
$$\cdots$$
$$I_n = (I_{nL}, I_{nR})$$

(2) A 用盲签名协议隐蔽所有 n 张汇票。并全部交给银行。

(3) 银行要求 A 恢复出随机的 n－1 张汇票,并确认它们都是合法的。银行检查总

数、唯一字符串并要求 A 出示所有认证字符串。

(4) 如果银行对 A 没有任何进行欺骗的企图感到满意,就在余下的一张隐蔽汇票上签名。银行把这张隐蔽汇票交回 A,并从她的账户上扣除这笔钱。

(5) A 恢复这张汇票,并在一个商人那里花掉它。

(6) 商人验证银行的签名以确认这张汇票是合法的。

(7) 商人要求 A 随机按汇票上每个认证字符串的左半或右半。实际上,商人给 A 一个随机的 n 比特选择字符串,$b_1, b_2, \cdots, b_n$。A 根据 $b_i$ 是 0 还是 1 公开 $I_i$ 的左半或右半。

(8) A 同意。

(9) 商人拿着这张汇票来到银行。

(10) 银行验证这个签名并检查它的数据以确认有相同唯一字符串的汇票先前没有存过。如果没有,银行把这笔钱划到商人的账上。银行在它的数据库中记下这个唯一字符串和所有识别信息。

(11) 如果这个唯一字符串在数据库中,银行就拒收汇票。接着,它把汇票上的识别字符串同它数据库中存的相比较。如果相同,银行知道是商人复制了汇票。如果不同,银行知道是买汇票的人影印了它。由于接收这张汇票的第二个商人交给 A 一个和第一个商人不同的选择字符串,银行找出一个比特位,在这个比特位上,一个商人让 A 公开了左半,而另一个商人让 A 公开了右半。银行异或这两半以揭露 A 的身份。

这是一个相当迷人的协议,下面让我们从不同角度来看看它。

A 能进行欺骗吗?她的数字汇票不过是一个比特字符串,所以可以复制它。第一次花它不会有问题,她只需完成协议,则一切进展顺利。商人在步骤(7)中给她一个随机的 n 比特选择字符串,并且 A 在步骤(8)中将公开每个 $I_i$ 的左半或右半。在步骤(10)中,银行将记录所有这些数据,连同汇票的唯一字符串。

当她试图第二次使用同一张数字汇票时,商人(同一个商人或另一商人)将在步骤(7)中给她一个不同的随机选择字符串。A 必须在步骤(8)中同意;不这样做势必立即提醒商人有些事值得怀疑。现在,当这个商人在步骤(10)中将汇票带到银行时,银行会立即发现带相同唯一字符串的汇票已经存过。银行接着比较认证字符串中所有公开的部分。两个随机选择字符串相同的机会是 $2^n$ 分之一,在下一个汇期前是不可能发生的。现在,银行找出这样一对,其中一半第一次公开,另一半第二次公开。它把这两半一起异或,马上得到 A 的名字,于是银行知道谁试图两次花这一张汇票。

应当指出,这个协议不能让 A 不进行欺骗,但它几乎肯定能检测到 A 的欺骗。如果 A 进行欺骗,她不可能不暴露身份。她不可能改变唯一字符串或任何识别字符串,因为此时银行的签名不再有效。这个商人将在步骤(6)中马上发现这点。

A 可能试图偷一张空头汇票骗过银行,这张汇票上的识别字符串不会泄露她的名字,或最好是一张其识别字符串泄露其他人名字的汇票。她在步骤(3)中进行这种欺诈骗过银行的机会是 n 分之一。这些并非是不可能的机会,但如果作出的惩罚足够严厉的话,A 不敢以身试法。或者,可以增加 A 在步骤(1)中制作的多余汇票的数目。

这个商人能进行欺骗吗?他的机会更小。他不能将这张汇票存两次;银行将会发现

选择字符串被重复使用。他不能捏造陷害 A，只有 A 才能打开任意识别字符串。

甚至 A 和商人合谋也不能欺骗银行。一旦银行在带唯一字符串的汇票上签名，银行就确认只能使用这张汇票一次。

银行又怎样呢？它能不能知道从商人那儿收到的汇票是它为 A 签的那张呢？在步骤(2)～步骤(5)中的盲签名协议保护了 A。银行无法作出判断，即使它保留了每次交易的完整记录，银行和商人在一起也无法知道 A 是谁。A 可以走进商店并且完全匿名地购买东西。

E 可以进行欺骗。如果她能窃听 A 和商人之间的通信，并能在商人到银行之前先到银行，她就能第一个把这笔数字现金存入她的账户。银行将会接受，甚至当商人试图去存入数字现金时他会被认为是一个欺骗者。如果 E 偷到数字现金并在 A 之前花掉它，那么 A 会被认为是一个欺骗者。没有办法防止这种情况；它是现金匿名的直接后果。当他们要使用纸币时，A 和商人都必须保护好他们的每一比特信息。

这个协议是介于被仲裁协议和一个自我执行协议之间的协议。A 和商人都相信银行能兑现汇票，但 A 不必信任知道她购物的银行。

### 6.3.6.5　数字现金与犯罪

数字现金也有它不利的一面。有时人们并不需要那么多的隐私。看看下面的例子中 A 进行的高明犯罪：

(1) A 绑架了一个婴儿。

(2) A 准备了 10 000 张匿名汇票，每张面额 1000 美元。

(3) A 用盲签名协议隐蔽所有 10 000 张汇票，她把它们送给当局并威胁除非按下列指示去做，否则要杀死婴儿：

① 让银行签所有的 10 000 张汇票。

② 在报纸上公布结果。

(4) 当局同意。

(5) A 买了一张报纸，恢复那些汇票，并开始花这些钱。当局没有办法靠追踪这些汇票来抓到她。

(6) A 放了这个婴儿。

注意：这种情况比任何涉及实际特征的情况更麻烦。因为缺乏物理接触，警察很难有机会抓住绑架者。

虽然如此，数字现金对犯罪分子来说也算不上理想。问题是匿名只有一种方式奏效：消费者是匿名而商人不是。而且，商人不能隐藏他收到钱的事实。

数字现金使政府容易知道某人挣了多少钱，但很难知道他把钱花在什么上。

### 6.3.6.6　其他数字现金协议

还有一些数字现金协议，涉及到相当复杂的数学问题。

通常，各种数字现金协议可以分为不同种类。在线式系统需要商人在每次销售时和银行联系，很像今天的信用卡协议。如果有问题，银行不会接受数字现金，A 不能行骗。

脱线式系统，如抗抵赖协议，直到商人与顾客交易之后它都不需要商人和银行之间的通信。这类系统可以发现 A 行骗但不能防止 A 行骗。抗抵赖协议中可以通过验证 A 的身份知道她是否试图欺骗来发现她的欺骗。A 知道这种情况会出现，因此她不会欺骗。

另一种方法是制造一个特定的智能卡，它包含一个叫做“观察者”的防篡改芯片。“观察者”芯片保存一个所有关于智能卡上花掉的数字现金信息的袖珍数据库。如果 A 企图复制数字现金并再次花掉它，这个嵌入在内的“观察者”芯片就会发现并禁止交易。因为“观察者”芯片是防篡改的，A 不能抹掉袖珍数据库，除非永久性的损坏智能卡。数字现金以它自己的方式在经济领域中流通；当它最终存入银行时，银行可以检查数字现金并发现是否有人进行欺骗以及谁在进行欺骗。

数字现金协议也可以被分在另一类。电子货币有固定的价值；使用这个系统的人们需要若干不同面额的硬币。电子支票可以用在任何数量直到最大值上，然后作为退款返回没有用完的部分。

Tatsuaki Okamoto 和 Kazuo Ohta 列出了一个理想数字现金系统的 6 个性质：

（1）独立性。数字现金的安全性不依赖于任何物理位置。现金能通过网络传送。

（2）安全性。数字现金不能复制和重用。

（3）隐私性（不可追踪性）。用户的隐私受到保护，没有人能追踪发现用户和他们所购物之间的关系。

（4）脱线付款。当一个用户用电子现金为所购物付款时，用户和商人之间的协议是脱线执行的。那就是说，商店不必与一台主机相连以处理用户的付款。

（5）可转移性。数字现金可转移给其他用户。

（6）可分性。给定数量的数字现金能分成较小数额的几份数字现金（当然，每份加起来总数还是那么多）。

上面讨论的协议满足性质（1）、（2）、（3）和（4），但不满足性质（5）和（6）。还有一些满足除性质（4）以外的所有性质的在线式数字现金系统。

#### 6.3.6.7 匿名信用卡

这个协议用在若干不同的银行以保护顾客的身份。每一位顾客在两个不同银行拥有一个账户，第一个银行知道此人的身份并愿意发给他信用卡，第二个银行仅仅知道他的假名（类似于瑞士银行的账号）。

顾客可以通过证明账户是他的，从第二个银行取出资金。但是，银行不知道这个人，也不愿发给他信用卡。第一个银行知道这个顾客并转账给第二个银行（用不着知道假名）。然后顾客就可以匿名地使用这笔资金。在月末，第二个银行给第一个银行一份账单，这是银行真正应该支付的。第一个银行把顾客应该支付的账单送给顾客。当顾客付账后，第一个银行把附加资金转账到第二个银行。

所有交易都是通过一个中间媒介处理的，就像电子联邦储备所做的那样：在银行之间结算账目，登记消息并产生审计追踪。

除非每个人都串通起来陷害顾客，顾客的匿名是可以保证的。然而，这不是数字现

金；银行很容易进行欺骗。这个协议使顾客在不泄露隐私的情况下保护其信用卡的利益。

# 6.4　公钥基础设施

公钥基础设施(public key infrastructure,PKI)是一个包括硬件、软件、人员、政策和手续的集合,实现了基于公钥密码体制的证书产生、管理存储、发行和作废等功能。

PKI 采用了证书管理公钥,通过第三方的可信任机构认证中心,把用户的公钥和用户的其他标识信息捆绑在一起,在 Internet 上验证用户身份。PKI 基础设施把公钥密码和对称密码结合起来,在 Internet 上实现密钥的自动管理,保证网上数据的安全传输。

PKI 工作组给 PKI 的定义是：一组建立在公开密钥算法基础上的硬件、软件、人员和应用程序的集合,它应具备产生、管理、存储、分发和废止证书的能力。一个典型的 PKI 体系结构包括认证中心 CA、注册机构 RA、证书持有者、应用程序、存储仓库 5 个组成部分。完整的 PKI 包括认证政策的制定、认证规则、运作制度的制定、所涉及的各方法律关系内容以及技术的实现。

## 6.4.1　PKI 的体系结构

由以下 5 个基本部分组成。

(1) 认证机构(certificate authority,CA)：是受信任的证书签发机构,是 PKI 体系的核心。CA 负责为 PKI 体系中所有实体发放证书,更新证书,废除发放的证书。

(2) 证书库：集中 CA 发布的所有证书。提供方便,高效的证书查询功能。方便用户查找其他用户证书。通常采用支持 LDAP 协议的目录服务系统来发布证书。

(3) 密钥备份及恢复系统：对于由 CA 代为生成的密钥对,可以在用户授权的情况下为用户备份密钥。在用户证书损坏或遗失时,可以给用户恢复密钥。密钥备份与恢复应该只针对解密密钥,而不能用在签名密钥。

(4) 证书作废处理系统：在用户证书遗失、泄密等情况下,需要证书作废系统及时把证书作废信息发布出去,把用户的损失减到最小。CA 一般通过发布证书废除列表(CRL)来发布作废信息。CRL 是由 CA 签名的一组电子文档,包括被废除证书的唯一标识(证书序列号)。

(5) 应用接口系统：PKI 的价值在于使用户能够方便使用加密、数字签名等安全服务,因此一个完整的 PKI 必须提供良好的应用接口系统,使得各种各样的应用能够以安全、一致、可信的方式与 PKI 交互,确保所建立起来的网络环境的可信性,同时降低管理维护成本。PKI 应用接口系统应该是跨平台的。

PKI 的主要目的是通过自动管理密钥和证书,为用户建立起一个安全的网络运行环境,使用户可以在多种应用环境下方便地使用加密和数字签名技术,从而保证网上数据的机密性、完整性、有效性。数据的机密性是指数据在传输过程中,不能被非授权者偷看；数据的完整性是指数据在传输过程中不能被非法篡改；数据的有效性是指数据不能被否认。

一个有效的PKI系统必须是安全的和透明的，用户在获得加密和数字签名服务时，不需要详细了解PKI是怎样管理证书和密钥的。一个典型、完整、有效的PKI应用系统必须能够实现以下功能：注册、发证、密钥恢复、密钥产生、密钥更新、交叉认证、证书废止。

使用基于公钥技术系统的用户建立安全通信信任机制的基础是：网上进行的任何需要安全服务的通信都是建立在公钥基础之上的，而与公钥成对的私钥只掌握在他们与之通信的另一方。这个信任的基础是通过公钥证书的使用来实现的。公钥证书就是一个用户的身份与他所持有的公钥的结合，在结合之前由一个可信任的权威认证机构CA来证实用户的身份，然后由其对该用户身份及对应公钥相结合的证书进行数字签名，以证明其证书的有效性。

## 6.4.2 PKI的基本内容

由第三方信托、认证权威和证书构成。

### 1. 第三方信托

第三方信托指的是在某种情形下，双方在毫无个人交往的情况下相互信任对方。两个陌生人如果同时和某个第三方有交往，而这第三方能担保这两个互不相识的人可信任，那么这两个陌生人就能相互信任了。要实现任何一个大规模的网络安全，第三方信托是必需的基础。在一大群人中建立第三方信托，就需要建立一个权威来确保每一方的可信性。在公钥加密技术中，第三方一半是人为的，另一半是自动的。一个受信赖的由人组成的权威机构必须作出谁可信赖谁不可信赖的决定。自动化则可以管理实际操作中的信任关系。

### 2. 认证权威

一个认证权威机构是一个受信托的组织，它的责任就是确认用户的真实性。CA的功能有些像一个政府的护照办公室。护照是一个公民的安全文件，由专门的权威机构颁发，护照能证实该公民的身份，是持照人的纸上身份。不论哪个国家，只要相信这个政府的发照机构，就会相信这个公民的护照。这就是第三方信托的最好例子。CA颁发的一个网络用户的电子身份就像是他的护照一样，能证明这个用户得到了这个CA的信任。不论谁只要相信这个CA，通过第三方信托，同样也相信这个用户。一个护照发放机构和一个CA都是政策和实体的结合。作为护照发放机构，政府订下政策，决定谁是公民，他们怎样才能得到护照。同样，也可以把CA想像成一个组织，这个组织既决定网络安全的政策，又决定谁能在这个组织的网络中得到电子认证。

### 3. 证书

一个网络用户的证书就和一个公民的护照一样，都含有可证明持有人身份的可靠信息。比如，一个确认证书包含持证人的姓名和确认的公钥。别人可以用这个公钥来验证该公钥拥有者的数字签名。另一种证书，叫加密证书，包含着拥有者的加密公钥。别人可

以用它来为发给拥有人的信加密。信任证书的中心问题就是证书里的信息是如何可靠保存的。一个人怎么才能相信证书里的姓名和公钥真的属于那个证书的拥有者？如果没有信任，公钥加密证书就不能成立。只有信任才能向人保证他们在向那个真正的持证人发送加密信件，或者确认数字签名真的是持证人的。为了把一个用户的公钥和其他信息组合在一起，并建立起可信性，CA 用自己的私有签名密钥给这个证书签名。CA 的数字签名给这个证书提供 3 个重要的安全和信任基础。

(1) 一个有效的数字签名是这个证书完整性的保障。

(2) 既然只有这个 CA 才能得到自己的签名私有密钥，那么，只要证实了证书上的 CA 签字，就可以确认只有这个 CA 才能产生这个签名。

(3) 既然只有这个 CA 才能得到自己的签名私有密钥，那么这个 CA 就不能否认自己在这个证书上的签名。CA 的许多应用软件和服务使它具有 PKI 的功能。在此情况下，他们只是发行证书，不能在用户密钥和证书的有效期内管理这些密钥和证书。这就导致了成本的增加和使用的困难。

在管理 PKI 中，证书和密钥在其有效期内都得到管理。密钥的作废和更新都是自动的。管理 PKI 提供这些功能确保 PKI 可靠，适应性强，易用，费用低，在现实网络安全解决方案中实际可行。

## 6.4.3 PKI 涉及的标准与协议

从整个 PKI 体系的建立与发展历程来看，与 PKI 相关的标准主要包括以下内容。

### 1. X.209(1988)ASN.1 基本编码规则的规范

ASN.1 是描述在网络上传输信息格式的标准方法。它有两部分：第一部分(ISO 8824/ITUX.208)描述信息内的数据、数据类型及序列格式，也就是数据的语法；第二部分(ISO 8825/ITUX.209)描述如何将各部分数据组成消息，也就是数据的基本编码规则。

ASN.1 原来是作为 X.409 的一部分而开发的，后来才独立成为一个标准。这两个协议除了在 PKI 体系中应用外，还广泛应用于通信和计算机的其他领域。

### 2. X.500(1993)信息技术之开放系统互联：概念、模型及服务简述

X.500 是一套已经被国际标准化组织(ISO)接受的目录服务系统标准，它定义了一个机构如何在全局范围内共享其名字和与之相关的对象。X.500 是层次性的，其中的管理域(机构、分支、部门和工作组)可以提供这些域内的用户和资源信息。在 PKI 体系中，X.500 用来唯一标识一个实体，该实体可以是机构、组织、个人或一台服务器。X.500 被认为是实现目录服务的最佳途径，但 X.500 的实现需要较大投资，并且比其他方式速度慢；而其优势具有信息模型、多功能和开放性。

### 3. X.509(1993)信息技术之开放系统互联：鉴别框架

X.509 是由国际电信联盟(ITU-T)制定的数字证书标准。在 X.500 确保用户名称

唯一性的基础上，X.509为X.500用户名称提供通信实体的鉴别机制，并规定实体鉴别过程中广泛适用的证书语法和数据接口。

X.509的最初版本公布于1988年。X.509证书由用户公共密钥和用户标识符组成。此外还包括版本号、证书序列号、CA标识符、签名算法标识、签发者名称、证书有效期等信息。这一标准的最新版本是X.509V3，它定义了包含扩展信息的数字证书。该版数字证书提供一个扩展信息字段，用来提供更多的灵活性及特殊应用环境下所需的信息传送。

#### 4. PKCS系列标准

由RSA实验室制定的PKCS系列标准，是一套针对PKI体系的加解密、签名、密钥交换、分发格式及行为标准，该标准目前已经成为PKI体系中不可缺少的一部分。

#### 5. OCSP在线证书状态协议

OCSP(online certificate status protocol)是IETF颁布的用于检查数字证书在某一交易时刻是否仍然有效的标准。该标准提供给PKI用户一条方便快捷的数字证书状态查询通道，使PKI体系能够更有效、更安全地在各个领域中广泛应用。

#### 6. LDAP轻量级目录访问协议

LDAP规范(RFC1487)简化了笨重的X.500目录访问协议，并且在功能性、数据表示、编码和传输方面都进行了相应的修改。1997年，LDAP第3版本成为互联网标准。目前，LDAPV3已经在PKI体系中被广泛应用于证书信息发布、CRL信息发布、CA政策以及与信息发布相关的各个方面。

除了以上协议外，还有一些构建在PKI体系上的应用协议，这些协议是PKI体系在应用和普及化方面的代表作，包括SET协议和SSL协议。

目前，PKI体系中已经包含众多的标准和标准协议，由于PKI技术不断进步和完善，以及其应用的不断普及，将来还会有更多标准和协议加入。

## 6.4.4 国外PKI/CA体系发展状况

作为电子商务信息安全的关键技术和基础技术的PKI/CA技术，对其体系的研究受到各国重视，从20世纪90年代初期以来，美国、加拿大、英国、德国、日本和新加坡等国相继开展了可信第三方认证体系的研究和建设工作。以美国、加拿大为例，通过对他们的体系结构的研究，可以从中得到一些启示。

### 6.4.4.1 美国联邦PKI体系结构

#### 1. 美国联邦PKI体系结构的描述

美国联邦PKI筹委会成立于1996年，由政府信息技术服务部、国家航空航天总署、国家标准技术研究所、国家安全部、国防部、交通部、财政部、农业部、劳动统计局和联邦网

络委员会等 20 个部、署共同组建而成。它与联邦首席信息官委员会共同致力于 FPKI 体系结构的研究。

联邦 PKI 支持在开放的网络如 Internet 上安全交易，用于保障电子政务、电子采购的信息安全和实现对关键网络设备的保护。特别是美国联邦 PKI 能帮助美国联邦机构与其他联邦机构，各级政府，贸易伙伴（私有性质的），公众机构之间进行电子交易。然而，美国联邦 PKI 并不是铁板一块，它是自下而上建立的一个庞大 PKI 体系。

联邦政府首先成功地在各联邦机构中分别使用了公开密钥密码技术，为网上交易提供全面解决方案，为所有服务颁发和管理数字证书：登记机构验证用户的身份，认证机构负责批准证书，颁发证书和发布证书黑名单，密钥恢复机构能恢复遗失的私钥等。PKI 产品和服务的多样性为机构的广泛用途提供支持，从而有效地提高了政府机构的工作效率，降低了办公成本。正是由于 PKI 产品和服务的多样性，有的机构或企业可能会购买 PKI 产品，运行他们自己的信任域；而有的机构或企业可能购买 PKI 服务，造成了彼此之间缺乏互操作性。为了增强国际间的合作，提高整体竞争力，解决不同信任域之间的互操作性问题，联邦政府计划联合各联邦机构中独立的 PKI/CA——美国国防部（department of defense，DoD），国家标准技术研究所（national institute of standardsand technology，NIST），乔治亚州技术研究所（georgia tech research institute，GTRI），州政府和地方政府等共同组建美国联邦 PKI 体系。使用该体系可以使信任域不仅局限在本信任域环境中，而且也可以使信任域扩展到整个联邦政府甚至是全球。

该体系结构主要由联邦桥认证机构（federal bridge CA，FBCA），首级认证机构（principal CA，PCA），次级认证机构（subordinate CA，SCA）等组成。其体系结构如图 6-6 所示。

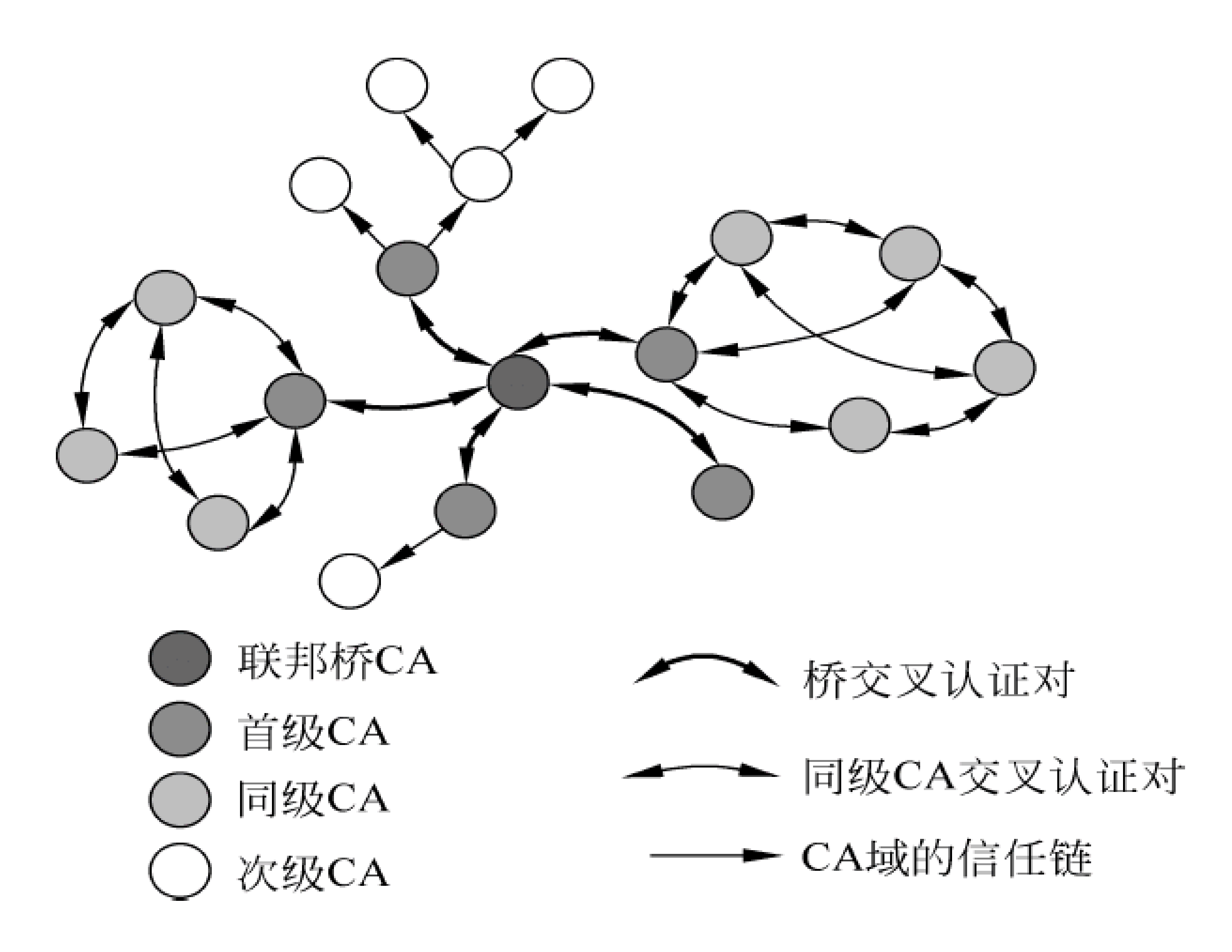



图 6-6　美国联邦 PKI 的体系结构

从图 6-6 中可以看到联邦 PKI 的体系结构中没有使用根 CA 的概念，取而代之的是首级 CA。这是因为在美国信任域的结构是多种多样的，联邦 PKI 体系结构可以支持分级（树状）结构、网状结构和信任列表等，而只有树状结构中的首级 CA 才称作根 CA。因此它允许加入联邦 PKI 体系中的机构使用任何结构的 PKI 信任域。联邦桥 CA 是联邦

PKI 体系中的核心组织，是不同信任域之间的桥梁 CA，主要负责为不同信任域的首级 CA 颁发交叉认证的证书，建立各个信任域的担保等级与联邦桥 CA 的担保等级之间的一一映射关系，更新交叉认证证书，发布交叉认证证书，注销黑名单。但是，它不要求一个机构在与另一个机构发生信任关系时必须遵循联邦 PKI 所确定的这种映射关系，而是可以采用它认为合适的映射关系确定彼此之间的信任。

### 2. 联邦 PKI 体系结构的工作原理

联邦 PKI 体系结构的工作原理实际上就是指联邦桥 CA 的工作原理。联邦桥 CA 不是一个树状结构的 CA，也不像网状 CA，它不直接向用户颁发证书；不像根 CA 一样成为一个信任点，它只是一个单独的 CA；它与不同的信任域之间建立对等的信任关系，允许用户保留他们自己的原始信任点。正如我们在网络中所使用的“HUB”一样，任何结构类型的 PKI 结构都可以通过这个机构连接在一起，实现彼此之间的信任，并将每一个单独的信任域通过联邦桥 PKI 扩展到整个联邦 PKI 体系中。

在进行网上交易时，当接收者接收到发送者数字签名的电子文件时，为了验证签名的有效性，接收者的应用软件必须做 3 件事：

(1) 接收者的软件必须确定发送者的信任域和接收者的信任域之间是否存在信任关系，这可以通过建立两个信任域之间的证书“信任路径”来实现。

(2) 接收者必须确定发送者证书中所描述的政策即证书包含的担保级别满足本次交易的需要。

(3) 确定在这个信任路径中，所有证书都必须是有效的，即它们既没有超过有效期，也没有被注销。如果接收者和发送者是在相同的信任域，以上 3 个步骤将简单明了地执行，因为，处于同一个信任域中的两个用户有相同的信任点和相同证书政策。如果接收者的信任域和发送者的信任域是不同的，这种情况就非常复杂，它可以借助联邦桥 CA 建立信任路径。在联邦桥 CA 与不同信任域中的首级 CA 进行交叉认证时，联邦桥 CA 颁发的证书中包含了联邦桥 CA 的担保等级与首级 CA 的担保等级之间的一一映射关系。借助这种映射关系，构建两个不同信任域之间的信任桥梁。

### 3. 联邦 PKI 体系的当前状况和联邦政府计划

从 2000 年 3 月 7 日至 4 月 6 日之间，在美国部分联邦机构的 CA 和加拿大政府 CA 的参与下，在有限时间内共同完成了对联邦 PKI 体系中的安全电子邮件数字签名的测试。其测试结果显示除 GSA 信任域外的其他信任域之间的互操作性是可以实现的，而使用 GSA 信任域不能实现与其他信任域的互操作性，报告同时强调，出现这种情况并不是技术问题，而仅仅是因为没有足够时间开发客户端验证软件。其进一步测试将继续进行。其测试结构图如图 6-7 所示，参加测试的联邦机构 CA 包括：DoD 的网状 CA，两个 NIST 的 CA，美国的国家航空和航天局(National Aeronautics and Space Administration, NASA)、乔治亚州技术研究所的 CA 及综合服务局(General Services Administration, GSA)的 CA 等。

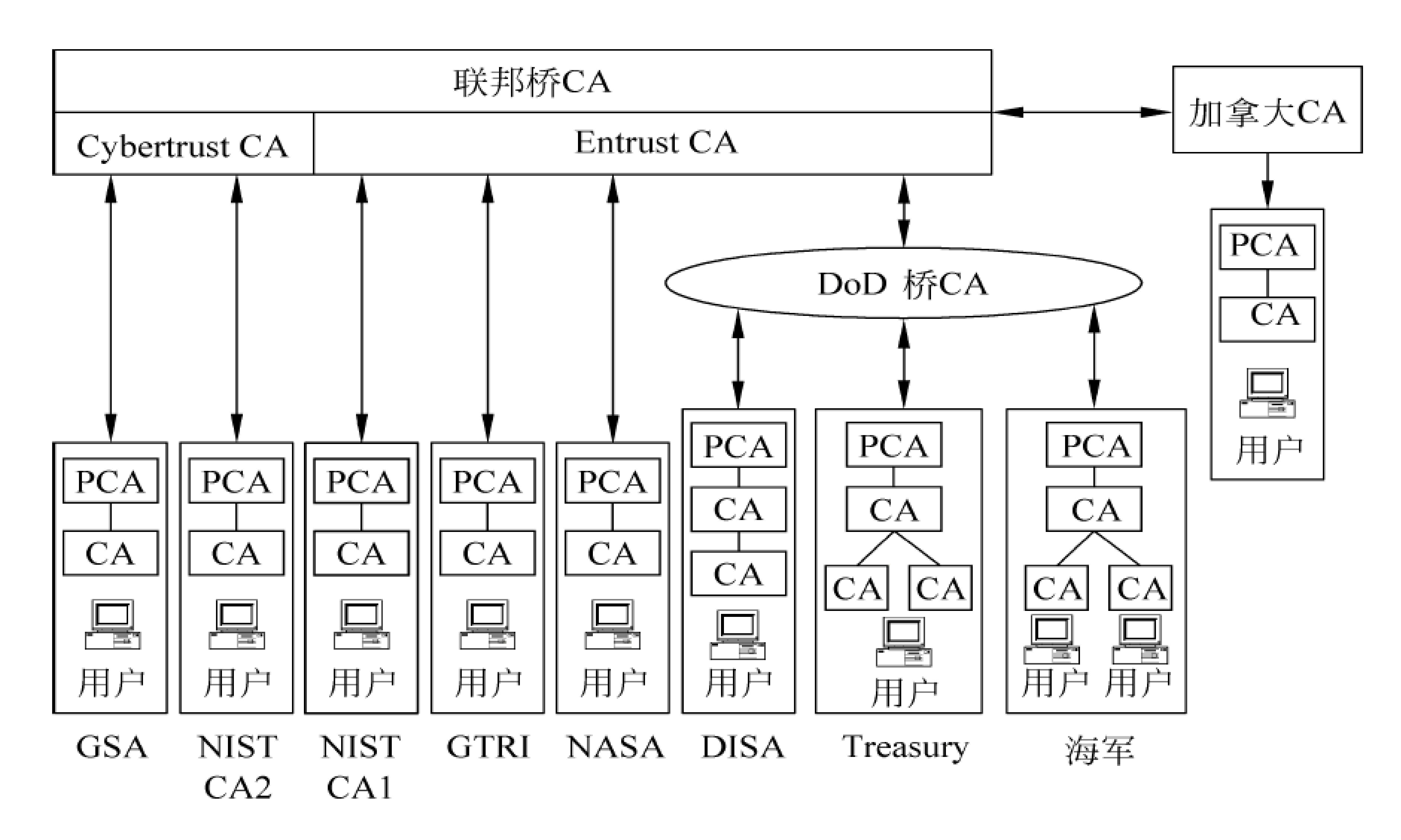



图 6-7　FPKI 体系测试结构图

图 6-7 中：

DISA：Defense Information Systems Agency，美国国防部国防信息系统局；

Treasury：美国国防部财政局；

PCA：首级 CA。

目前，美国政府联邦 PKI 体系还处于研究和测试阶段，正式投入使用还需要做很多工作。测试基本成功只是建立联邦 PKI 体系的第一步，为了建立健全联邦 PKI 体系，从技术角度，美国联邦政府还将从以下 4 个方面作出努力：

（1）验证密钥管理证书的信任路径。

（2）测试附加功能，以满足联邦 PKI 体系中的所有成员之间的互操作性。

（3）开发 FBCA 产品，FBCA 的设计依赖 X.509 证书的政策和转换政策信息的政策映射扩展项，这些特点将包含在 FBCA 产品中，同时开发的 FBCA 产品能够处理证书中的混合签名算法。

（4）努力加强在现有商务软件（commercial off the shelf-software，COTS）产品中 PKI 的支持。

### 6.4.4.2　加拿大政府 PKI 体系结构

#### 1. 加拿大政府 PKI 体系结构的描述

加拿大对于政府 PKI 体系的研究要比美国早，在 1993 年加拿大通信安全部（Communications Security Establishment，CSE）就已经开始了政府 PKI 体系雏形的研究工作，当时主要是开发一种满足政府需求的 PKI 产品，实现无货架商业贸易。随后，陆续有部分联邦政府机构参与了 GOCPKI 体系的开发工作，他们是：加拿大公民移民局（Citizenship and Immigration Canada），加拿大外事和国际贸易部（Department of Foreign Affairs and International Trade），国防部（Department of National Defence），加

拿大电信和信息服务中心(Government Telecommunications and Informatics Services),加拿大公共建设工程和政府服务中心(Public Works and Government Services Canada,PWGSC)的一个分部,加拿大卫生局(Health Canada),加拿大皇家骑警(Royal Canadian Mounted Police)和财政部秘书处(Treasury Board Secretariat)等。经过若干年的研究,2000年在建立开放的PKI体系方面获得重要的进展,政府PKI体系为联邦政府与公众机构,商业机构等进行电子数据交换时提供信息安全的保障,从而推动政府内部管理电子化的进程(其结构如图6-8所示)。

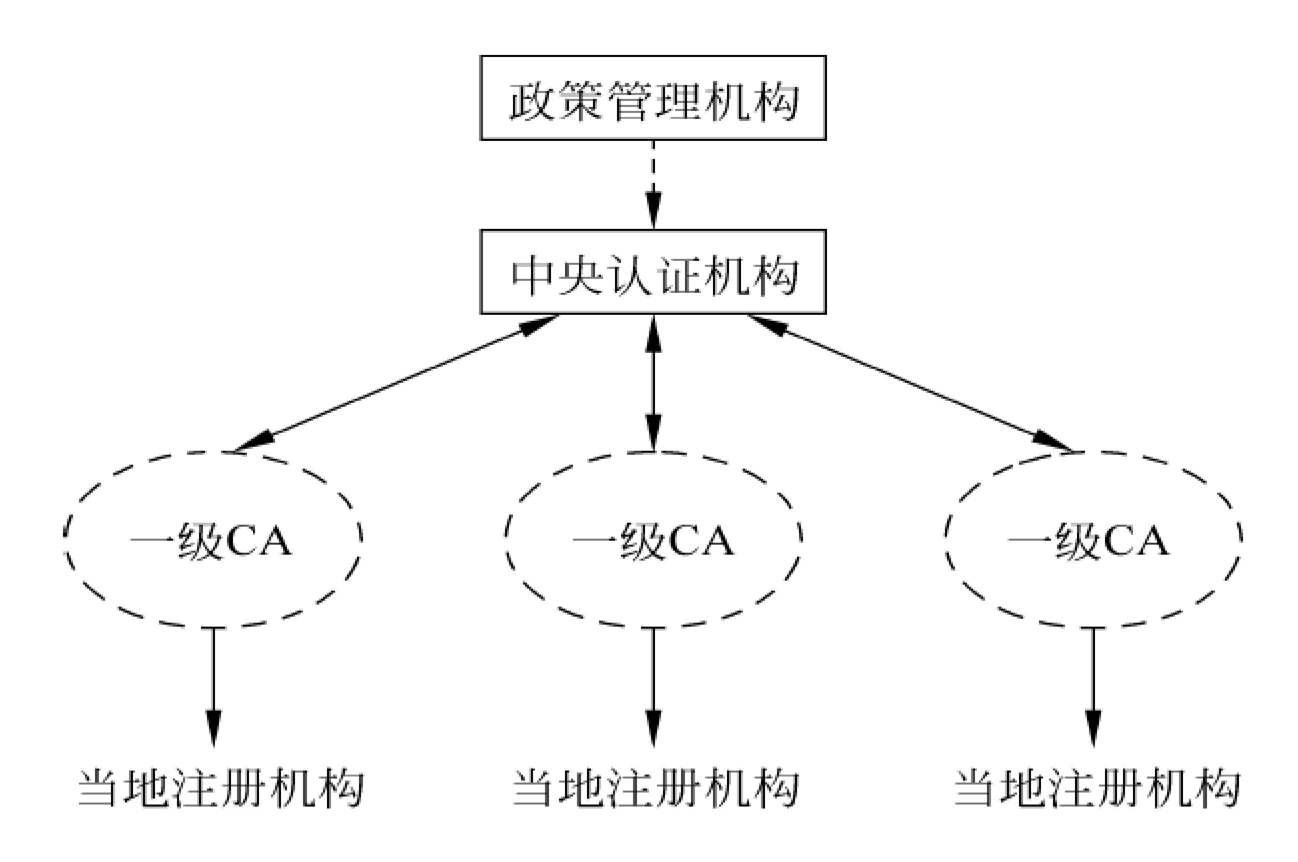



图 6-8 加拿大政府 PKI 体系结构

从图6-8中可以看到,加拿大政府PKI体系结构是由政策管理机构(PMA)、中央认证机构(CCF)、一级CA和当地注册机构(LRA)组成。

其中,PMA是一个若干部门共同组建的机构,由加拿大政府财政部秘书处领导,为政府PKI体系提供全面的政策指导,负责监督和管理加拿大政府PKI体系的政策实施情况。

CCF是中央认证机构,它实施政府PKI体系中的所有策略,签署和管理与一级CA交叉认证的证书。

一级CA是由政府运营,制定一个和多个证书担保的等级,分发和管理数字证书,定期颁布证书,注销黑名单。

当地注册机构(LRA)是一级CA设置的登记机构或个人,其职责是认证和鉴别申请者的身份;为密钥恢复或证书恢复请求进行审批;接受并审批证书的注销请求。

加拿大政府PKI体系是一个完全政府行为的公开密钥体系结构,它充分考虑到交易的私有性和安全性,并把保护交易私有性和安全性列为“信息高速公路”的首要问题。它提供完全一致的密钥管理办法,并为加密和数字签名提供完全相同的验证过程,它是多项技术——电子认证、鉴别和智能卡等的集成。

**2. 加拿大政府 PKI 体系的工作原理**

加拿大政府PKI体系是由若干CA组成,这些CA形成树状结构,每个用户的公钥和身份验证的信息都放在证书中,证书签发CA在每个证书上签名,并使其包含公钥的证书

对外公开。任何用户都能方便地得到其他用户的公钥,通过公钥验证该用户的签名,辨别真伪。

加拿大政府 PKI 体系的工作原理与美国联邦 PKI 体系的工作原理比较相似,需要建立信任关系的两个用户如果属于同一个树状结构,由于有共同的信任点,应用软件很容易验证他们的信任关系;如果两个用户不属于同一个树状结构,这时要建立信任关系必须借助中央认证机构的交叉认证机制。通过中央认证机构与一级 CA 之间的交叉认证证书,分属不同树状结构的一级 CA 之间建立彼此的信任关系。同样,中央认证机构也不是一个根 CA,它只是不同树状信任域的汇结点。

然而,从图 6-8 中也很容易发现,加拿大政府 PKI 体系的信任域都是树状结构,从而信任关系的查找更加快捷,信任关系很容易建立,只需要和相应的一级 CA 进行交叉认证即可,不像网状结构,需要确定哪个 CA 与联邦桥 CA 进行交叉认证。另外,两种树状结构建立信任关系必须通过中央认证机构,不允许一个树状结构与另一个树状结构直接发生信任关系,中央认证机构是与外界建立信任关系的唯一接口,因此结构简单,易于操作,是一个值得借鉴的 PKI 体系。

#### 6.4.4.3　两种体系的比较

美国联邦 PKI 体系和加拿大政府 PKI 体系都是政府行为的 PKI 体系,是在政府的倡导和主持下研究开发的,其目的都是为了本国的各级政府部门和政府机构高效、低成本、安全地从事电子政务、电子采购活动,其成员主要是各级政府,不同的政府机构,在两种体系中均有一个交叉认证中心(FPKI 体系中是联邦桥 CA,加拿大政府 PKI 体系中的中央认证机构),由它来沟通不同信任域之间的信任关系,同时也是与其他政府 PKI/CA 建立信任关系的接口,是该体系与外界建立信任关系的唯一通道,当然它必须接受政府在政策上的支持和监督,在技术上都集成了如智能卡技术等其他技术。美国在开发联邦 PKI 体系时充分考虑了与加拿大政府 PKI 体系的兼容性。然而美国联邦 PKI 体系和加拿大政府 PKI 体系并不完全一致,他们都有各自的特点,其表现如下:

(1) 体系结构上,美国联邦 PKI 体系结构比较复杂,它包含各种信任域结构:树状结构、网状结构和信任列表等。因此,联邦桥 CA 仅是一个桥梁;而加拿大政府 PKI 体系结构比较简单,它是一个树状结构。从结构上看,中央认证机构仿佛是一个根 CA。

(2) 在信任关系的建立上美国联邦 PKI 体系结构中的联邦桥 CA 是各信任域建立信任关系的桥梁,然而他并不强调在建立信任关系时必须遵循交叉认证证书中所确定的担保等级之间的一一映射关系;在加拿大政府 PKI 体系中,各信任域之间建立信任关系必须经过中央认证机构。

(3) 采用技术上,美国联邦 PKI 体系中的成员采用多种不同 PKI 产品和技术,如 VeriSign、Baltimore 和 Entrust 等公司的技术;而加拿大政府 PKI 体系中强调使用 Entrust 公司的技术。

(4) 在组成成员上,美国联邦 PKI 体系中除了各级政府和不同政府机构外,还包括与政府或政府机构有商业往来的合作伙伴;而加拿大政府 PKI 体系中的成员都是联邦的各级政府或政府机构。

### 6.4.5 国内 PKI 应用状况

目前，上海、北京、深圳、重庆等城市已经建立了 PKI，以便为本地化通信网络提供安全服务。

因为 CA 认证中心是 PKI 的重要组成部分，有时将 CA 认证中心称作简要的 PKI。在国家直属部门，以中国人民银行为首的 12 家金融机构推出了"中国金融认证中心(CFCA)"，中国电信也在开展 CA 机制的试验工作。另外，许多网络通信公司正在积极开发自己的基于 PKI 的安全产品。

但是，在我国，各地方推出的 PKI 基本上是为了满足局部需要，全国还没有由政府或金融机构推出的统一信任认证机制。

## 6.5 防火墙技术中安全协议的应用

防火墙是设置在被保护网络和外部网络之间的一道屏障，以防止发生不可预测的、潜在破坏性的侵入。防火墙是指设置在不同网络（如可信任的企业内部网和不可信的公共网）或网络安全域之间的一系列部件的组合。它是不同网络或网络安全域之间信息的唯一出入口，能根据企业的安全政策控制（允许、拒绝、监测）出入网络的信息流，且本身具有较强的抗攻击能力。它是提供信息安全服务，实现网络和信息安全的基础设施。

防火墙可通过监测、限制、更改跨越防火墙的数据流，尽可能对外部屏蔽网络内部的信息、结构和运行状况，以此来实现网络的安全保护。

### 6.5.1 防火墙的实质

防火墙包含一对策略（或称机制）：一方面它限制数据流通，另一方面又允许数据流通。由于网络的管理机制及安全策略（security policy）不同，因此这对矛盾呈现出不同的表现形式。

存在两种极端的策略：第一种是除了非允许不可的都被禁止，第二种是除了非禁止不可的都被允许。第一种的特点是安全但不好用，第二种是好用但不安全，而多数防火墙都在两者之间采取折中。

这里所谓的好用或不好用主要指跨越防火墙的访问效率。在确保防火墙安全或比较安全前提下提高访问效率是防火墙技术研究和实现的热点。

### 6.5.2 防火墙的技术分类

防火墙技术可根据防范的方式和侧重点的不同而分为很多种类型，但总体来讲可分为包过滤、应用级网关和代理服务器等几大类型。

#### 1. 数据包过滤型防火墙

数据包过滤（packet filtering）技术是在网络层对数据包进行选择，选择的依据是系统内设置的过滤逻辑，被称为访问控制表（access control table）。通过检查数据流中每个数

据包的源地址、目的地址、所用的端口号、协议状态等因素或它们的组合来确定是否允许该数据包通过。

数据包过滤防火墙逻辑简单，价格便宜，易于安装和使用，网络性能和透明性好，它通常安装在路由器上。路由器是内部网络与 Internet 连接必不可少的设备，因此在原有网络上增加这样的防火墙几乎不需要任何额外的费用。

数据包过滤防火墙有两个缺点：一是非法访问一旦突破防火墙，即可对主机上的软件和配置漏洞进行攻击；二是数据包的源地址、目的地址以及 IP 的端口号都在数据包的头部，很有可能被窃听或假冒。

分组过滤或包过滤，是一种通用、廉价、有效的安全手段。之所以通用，是因为它不针对各个具体的网络服务采取特殊的处理方式；之所以廉价，是因为大多数路由器都提供分组过滤功能；之所以有效，是因为它能在很大程度上满足企业的安全要求。所根据的信息来源于 IP、TCP 或 UDP 包头。

包过滤的优点是不用改动客户机和主机上的应用程序，因为它工作在网络层和传输层，与应用层无关。但其弱点也是明显的：据以过滤判别的只有网络层和传输层的有限信息，因而各种安全要求不可能充分满足；在许多过滤器中，过滤规则的数目是有限制的，且随着规则数目的增加，性能会受到很大影响；由于缺少上下文关联信息，不能有效地过滤如 UDP、RPC 一类的协议。另外，大多数过滤器中缺少审计和报警机制，且管理方式和用户界面较差；对安全管理人员素质要求高，建立安全规则时，必须对协议本身及其在不同应用程序中的作用有较深入的理解。因此，过滤器通常是和应用网关配合使用，共同组成防火墙系统。

### 2. 应用级网关型防火墙

应用级网关(application level gateways)是在网络应用层上建立协议过滤和转发功能。它针对特定的网络应用服务协议使用指定的数据过滤逻辑，并在过滤的同时，对数据包进行必要的分析、登记和统计，形成报告。实际中的应用网关通常安装在专用工作站系统上。

数据包过滤和应用网关防火墙有一个共同的特点，就是它们仅仅依靠特定的逻辑判定是否允许数据包通过。一旦满足逻辑，则防火墙内外的计算机系统建立直接联系，防火墙外部的用户便有可能直接了解防火墙内部的网络结构和运行状态，这有利于实施非法访问和攻击。

### 3. 代理服务型防火墙

代理服务(proxy service)也称链路级网关(circuit level gateways)或 TCP 通道(TCP tunnels)，也有人将它归于应用级网关一类。它是针对数据包过滤和应用网关技术存在的缺点而引入的防火墙技术，其特点是将所有跨越防火墙的网络通信链路分为两段。防火墙内外计算机系统间应用层的“链接”，由两个终止代理服务器上的“链接”来实现，外部计算机的网络链路只能到达代理服务器，从而起到了隔离防火墙内外计算机系统的作用。

此外，代理服务也对过往的数据包进行分析、注册登记，形成报告，同时，当发现被攻

击迹象时会向网络管理员发出警报,并保留攻击痕迹。

应用代理型防火墙是内部网与外部网的隔离点,起着监视和隔绝应用层通信流的作用。同时,也常结合过滤器的功能。它工作在 OSI 模型的最高层,掌握着应用系统中可用作安全决策的全部信息。

#### 4. 复合型防火墙

由于对更高安全性的要求,常把基于包过滤的方法与基于应用代理的方法结合起来,形成复合型防火墙产品。这种结合通常有以下两种方案。屏蔽主机防火墙体系结构:在该结构中,分组过滤路由器或防火墙与 Internet 相连,同时一个堡垒机安装在内部网络,通过在分组过滤路由器或防火墙上过滤规则的设置,使堡垒机成为 Internet 上其他节点所能到达的唯一节点,这确保了内部网络不受未授权外部用户的攻击。

屏蔽子网防火墙体系结构:堡垒机放在一个子网内,形成非军事化区,两个分组过滤路由器放在这一子网的两端,使这一子网与 Internet 及内部网络分离。在屏蔽子网防火墙体系结构中,堡垒主机和分组过滤路由器共同构成了整个防火墙的安全基础。

### 6.5.3 防火墙主要技术

先进的防火墙产品将网关与安全系统合二为一,具有以下技术与功能。

#### 1. 双端口或三端口的结构

防火墙产品具有两个或三个独立的网卡,内外两个网卡可不作 IP 转化而串接于内部网与外部网之间,另一个网卡可专用于对服务器的安全保护。

#### 2. 透明的访问方式

以前的防火墙在访问方式上要么要求用户作系统登录,要么需要通过 SOCKS 等库路径修改客户机的应用。新一代防火墙利用透明的代理系统技术,从而降低系统登录固有的安全风险和出错概率。

#### 3. 灵活的代理系统

代理系统是一种将信息从防火墙的一侧传送到另一侧的软件模块。新一代防火墙采用两种代理机制,一种用于代理从内部网络到外部网络的连接,另一种用于代理从外部网络到内部网络的连接。前者采用网络地址转换(NAT)技术来解决,后者采用非保密的用户定制代理或保密的代理系统技术来解决。

#### 4. 多级的过滤技术

为保证系统的安全性和防护水平,新型防火墙采用 3 级过滤措施,并辅以鉴别手段。在分组过滤一级,能过滤掉所有源路由分组和假冒 IP 源地址;在应用级网关一级,能利用 FTP、SMTP 等各种网关,控制和监测 Internet 提供的所用通用服务;在链路网关一级,实现内部主机与外部站点的透明连接,并对服务的通行实行严格控制。

5. 网络地址转换技术(NAT)

防火墙利用 NAT 技术能透明地对所有内部地址作转换,使外部网络无法了解内部网络的内部结构,同时允许内部网络使用自己定制的 IP 地址和专用网络,防火墙能详尽记录每一个主机的通信,确保每个分组送往正确的地址。同时使用 NAT 的网络,与外部网络的连接只能由内部网络发起,极大地提高了内部网络的安全性。NAT 的另一个显而易见的用途是解决 IP 地址匮乏问题。

6. Internet 网关技术

由于是直接串联在网络中,新一代防火墙必须支持用户在 Internet 互联的所有服务,同时还要防止与 Internet 服务有关的安全漏洞。所以它能以多种安全的应用服务器(包括 FTP、Finger、E-mail、Ident、News、WWW 等)来实现网关功能。为确保服务器的安全性,对所有文件和命令均要利用“改变根系统调用(ch root)”作物理上的隔离。

在域名服务方面,新一代防火墙采用两种独立的域名服务器,一种是内部 DNS 服务器,主要处理内部网络的 DNS 信息;另一种是外部 DNS 服务器,专门用于处理机构内部向 Internet 提供的部分 DNS 信息。

在匿名 FTP 方面,服务器只提供对有限受保护的部分目录的只读访问。在 WWW 服务器中,只支持静态网页,而不允许图形或 CGI 代码等在防火墙内运行;在 Finger 服务器中,对外部访问,防火墙只提供可由内部用户配置的基本文本信息,而不提供任何与攻击有关的系统信息;SMTP 与 POP 邮件服务器要对所有进、出防火墙的邮件做处理,并利用邮件映射与标头剥除的方法隐除内部的邮件环境;Ident 服务器对用户连接的识别作专门处理;网络新闻服务则为接收来自 ISP 的新闻开设专门磁盘空间。

7. 安全服务器网络(SSN)

为适应越来越多的用户向 Internet 上提供服务时对服务器保护的需要,新一代防火墙采用分别保护的策略,对用户上网的对外服务器实施保护,它利用一张网卡将对外服务器作为一个独立网络处理,对外服务器既是内部网的一部分,又与内部网关完全隔离。这就是安全服务器网络(SSN)技术,对 SSN 上的主机既可单独管理,也可设置成通过 FTP、Telnet 等方式从内部网上管理。

SSN 方法提供的安全性要比传统的“隔离区(DMZ)”方法好得多,因为 SSN 与外部网之间有防火墙保护,SSN 与内部网之间也有防火墙保护,而 DMZ 只是一种在内、外部网络网关之间存在的一种防火墙方式。换而言之,一旦 SSN 受破坏,内部网络仍会处于防火墙的保护之下;而一旦 DMZ 受到破坏,内部网络便暴露于攻击之下。

8. 用户鉴别与加密

为了降低防火墙产品在 Telnet、FTP 等服务和远程管理上的安全风险,鉴别功能必不可少。新一代防火墙采用一次性使用的口令字系统来作为用户的鉴别手段,并实现对邮件的加密。

### 9. 用户定制服务

为满足特定用户的特定需求,新一代防火墙在提供众多服务的同时,还为用户定制提供支持。这类选项有:通用 TCP,出站 UDP、FTP、SMTP 等。如果某一用户需要建立一个数据库的代理,便可利用这些支持,方便设置。

### 10. 审计和告警

新一代防火墙产品的审计和告警功能十分健全。日志文件包括:一般信息、内核信息、核心信息、接收邮件、邮件路径、发送邮件、已收消息、已发消息、连接需求、已鉴别的访问、告警条件、管理日志、进站代理、FTP 代理、出站代理、邮件服务器、名服务器等。告警功能会守住每一个 TCP 或 UDP 探寻,并能以发出邮件、声响等多种方式报警。此外,新一代防火墙还在网络诊断、数据备份与保全等方面具有特色。

## 6.5.4 设置防火墙的要素

### 1. 网络策略

影响 Firewall 系统设计、安装和使用的网络策略可分为两级,高级的网络策略定义允许和禁止的服务以及如何使用服务;低级的网络策略描述 Firewall 如何限制和过滤在高级策略中定义的服务。

### 2. 服务访问策略

服务访问策略集中在 Internet 访问服务以及外部网络访问(如拨入策略、SLIP/PPP 连接等)。服务访问策略必须是可行的和合理的。可行的策略必须在阻止已知的网络风险和提供用户服务之间获得平衡。典型的服务访问策略是:允许通过增强认证的用户在必要的情况下从 Internet 访问某些内部主机和服务;允许内部用户访问指定的 Internet 主机和服务。

### 3. 防火墙设计策略

防火墙设计策略基于特定的 Firewall,定义完成服务访问策略的规则。通常有两种基本设计策略:允许任何服务,除非被明确禁止;禁止任何服务,除非被明确允许。第一种特点是安全但不好用,第二种是好用但不安全。通常采用第二种类型的设计策略,而多数防火墙都在两种之间采取折中。

### 4. 增强的认证

许多在 Internet 上发生的入侵事件源于脆弱的传统用户/口令机制。多年来,用户被告知使用难于猜测和破译的口令,虽然如此,攻击者仍然在 Internet 上监视传输的口令明文,使传统的口令机制形同虚设。增强的认证机制包含智能卡、认证令牌、生理特征(指纹)以及基于软件(RSA)等技术,来克服传统口令的弱点。虽然存在多种认证技术,它们

均使用增强的认证机制，产生难被攻击者重用的口令和密钥。目前许多流行的增强机制使用一次有效的口令和密钥(如 SmartCard 和认证令牌)。

### 6.5.5　防火墙的抗攻击能力和局限性

作为一种安全防护设备，防火墙在网络中自然是众多攻击者的目标，故抗攻击能力也是防火墙的必备功能。

对防火墙的网络攻击手段一般包括 IP 地址假冒攻击、病毒攻击、口令字探询攻击、网络安全性分析攻击、邮件诈骗攻击等。

防火墙的局限性是：尽管利用防火墙可以保护安全网免受外部黑客的攻击，但其目的只是能够提高网络的安全性，不可能保证网络绝对安全。事实上仍然存在着一些防火墙不能防范的安全威胁，如防火墙不能防范绕过防火墙的攻击。例如，如果允许从受保护的网络内部向外拨号，一些用户就可能形成与 Internet 的直接连接。另外，防火墙很难防范来自于网络内部的攻击以及病毒的威胁。

## 6.6　VPN 技术中安全协议的应用

VPN(vitual private network)即虚拟专用网，是利用开放的公众网络资源建立私有数据传输通道，将远程的分支机构、商业伙伴、移动办公人员等连接起来，并且提供安全的端到端的数据通信的一种网络技术。VPN 有两层含义：它是“虚拟的”，即建立隧道或虚电路把不同的物理网络或设备连接起来，不再使用物理的专线建立专用网，而是将其建立在分布广泛的公共网络上，如 Internet；它是“专用的”，对基于 IPSec 的 VPN 而言，是一组连接的闭合用户群(CUG)，它不仅具有服务质量(QoS)的保证，而且更多地强调安全服务。VPN 所实现的用户之间的“端到端”的连接，实际上是 ISP(Internet 服务提供商)和其他 NSP(网络服务提供商)所提供的一种面向企业服务的增值业务。VPN 是企业网在公共网络上的无缝延伸，VPN 可将位于不同地点的远程用户、分支机构和合作伙伴等连接起来。VPN 拥有成本低、开销少、灵活度高等优点。

VPN 的主要功能是：

(1) 可以替换现有的专用网网段或子网。

(2) 通过把特定应用分离出来满足相应需求，为专用网络提供有益的补充。

(3) 在不影响现有专用网的情况下，处理新应用。

(4) 增加新位置，特别是国际性网站。

### 6.6.1　VPN 的基本原理

虚拟专用网系统使分布在不同地方的专用网络在不可信任的公共网络(如因特网)上安全地通信。它采用复杂的算法来加密传输的信息，使得需要受保护的数据不会被窃取。一般来说，其工作流程大致如下：

(1) 要保护的主机发送不加密信息到连接公共网络的虚拟专网设备。

(2) 后者根据网络管理员设置的规则，确认是否需要对数据进行加密或让数据直接

通过。

(3) 对需要加密的数据,虚拟专网设备对整个数据包(包括要传送的数据、发送端和接收端的 IP 地址)进行加密和附上数字签名。

(4) 虚拟专网设备加上新的数据包头,其中包括目的地虚拟专网设备需要的安全信息和一些初始化参数。

(5) 虚拟专网设备对加密后数据、鉴别包以及源 IP 地址、目标虚拟专网设备 IP 地址进行重新封装,重新封装后数据包通过虚拟通道在公网上传输。

(6) 当数据包到达目标虚拟专网设备时,数字签名被核对无误后数据包被解密。

VPN 的互联模型有两种,如图 6-9 和图 6-10 所示。

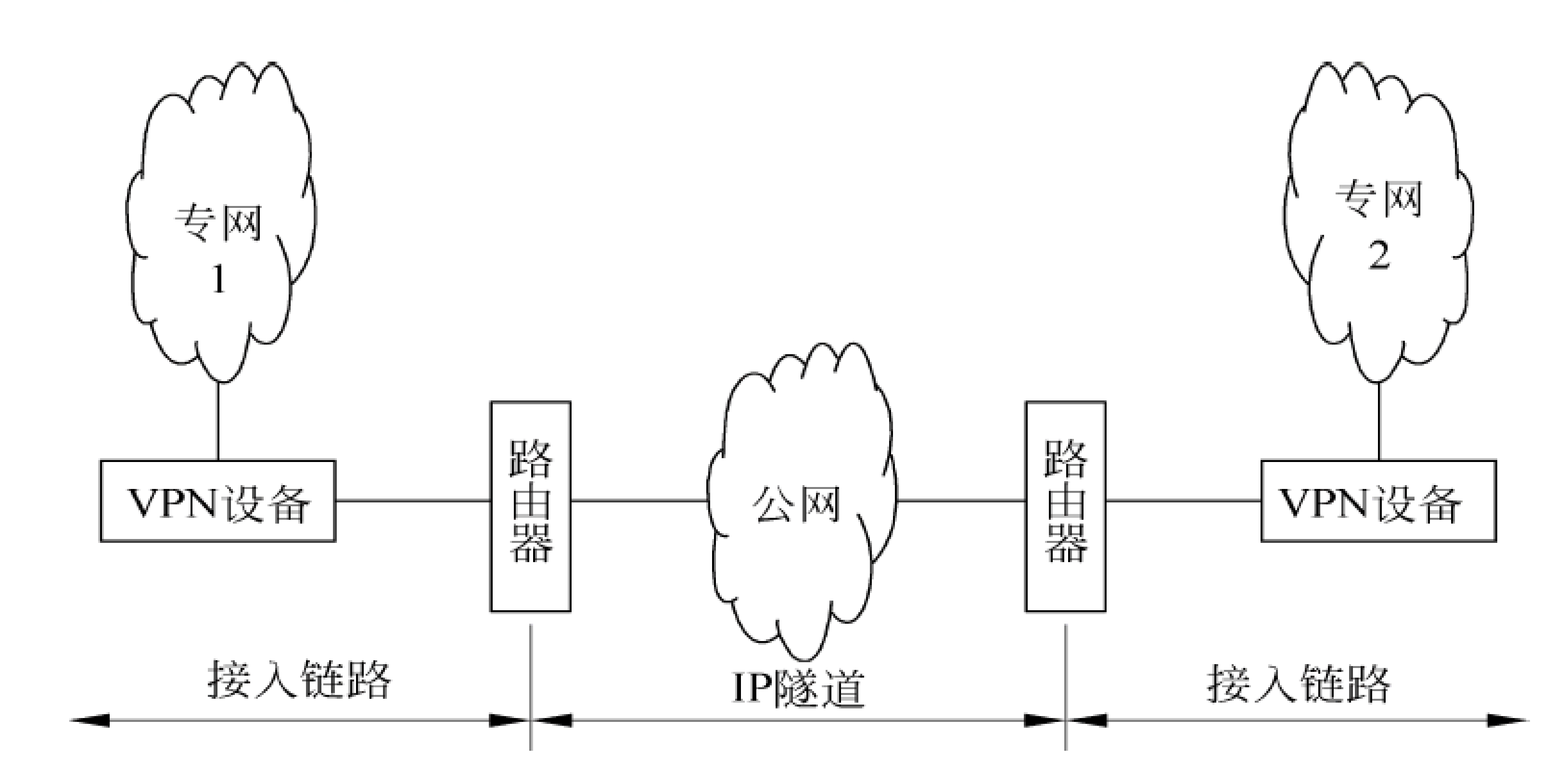



图 6-9 VPN 互联模型 1——用户网络与用户网络之间通过 VPN 的互联

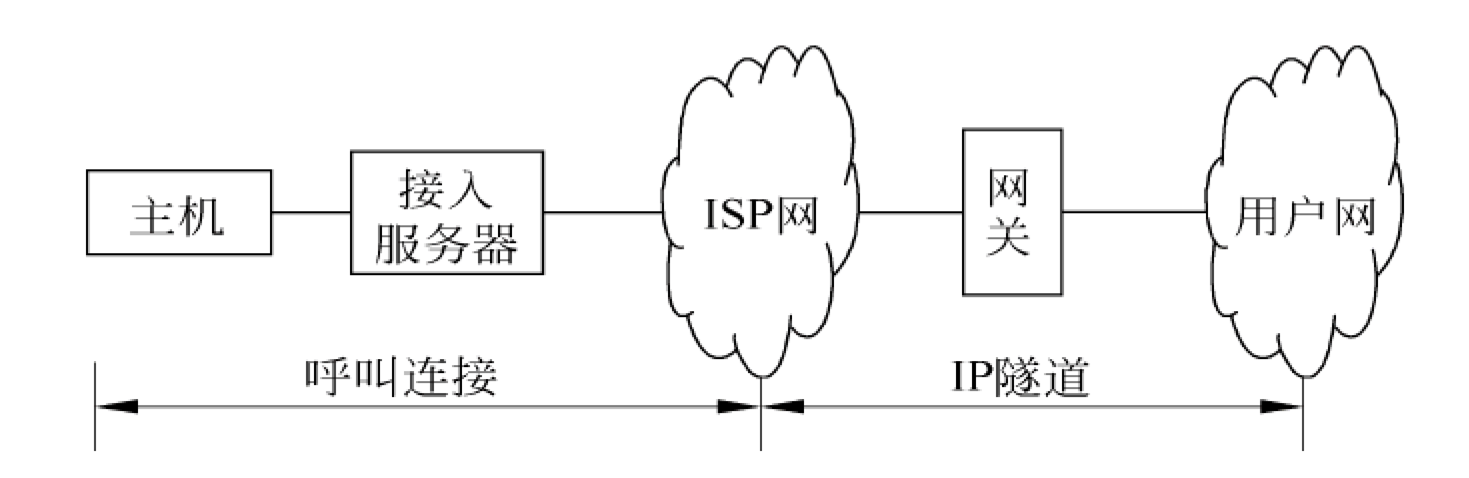



图 6-10 VPN 互联模型 2——主机与用户网络之间通过 VPN 的互联

## 6.6.2 VPN 采用的主要技术

虚拟专用网的具体实现是采用隧道技术,VPN 用户的数据封装在隧道中,在互联网中进行传输。隧道技术与客户接入方式无关,它可支持各种形式的接入,如拨号方式接入、Cable Modem、XDSL、综合业务数字网(ISDN)、El 专线、无线接入、以太网接入等。一个隧道协议通常包括以下几个方面:

(1) 乘客协议:被封装的协议,如 PPP、SLIP。

(2) 封装协议:隧道的建立、维持和断开,如 L2TP、IPSec 等。

(3) 承载协议:承载经过封装后的数据包的协议,如 IP 和 ATM 等。

VPN 的总体要求是:VPN 技术应当能够支持全网唯一的,能够辨别 VPN 的标识符

(VPN-ID)确定 VPN 的成员(VPN 边缘路由器需要从本地 Stub 链路来获取 VPN 成员信息),同时也要从属于同一 VPN 的其他路由器上获取 VPN 成员信息,这一过程实际是由以下两部分构成。

(1) Stub 链路连接信息的获得。边缘路由器必须通过每个 Stub 链路来获取与之相连的 CPE 的地址和地址前缀。

(2) VPN 内可到达信息的获取。边缘路由器得到了 Stub 链路连接信息(以及相连的 CPE 设备的地址前缀)之后,还必须将这些信息转发给同一 VPN 内的其他 VPN 边缘路由器。

隧道机制是边缘路由器必须要同 VPN 域内的其他路由器建立起必要的隧道以及隧道网络,在隧道内通过封装和解封操作对数据包进行发送和接收。

实际网络环境中,经常存在在一个用户端点通过多条 Stub 链路连接多台 VPN 边缘路由器的现象。这种现象主要分为两类:冗余链路和负载分担。

(1) 冗余链路:用户端路由器在同一时刻只支持一条链路实际工作,而其他链路则处于备份状态。

(2) 负载分担:用户端的一台路由器可以同时连接多台 VPN 边缘路由器,此时需要使用某种负载分担算法来防止循环的产生。

从总体来说,VPN 技术非常复杂,它涉及到通信技术、密码技术和现代认证技术,是一项交叉科学。目前,VPN 主要包含隧道技术、MPLS 技术与安全技术。

## 6.7　本章重点和难点

本章的重点是现实中使用的部分安全协议,由于这些协议与具体应用环境相关,所以在讲授时需要注意应用环境。

本科教学建议讲授 6.1 节、6.4.1 节、6.4.2 节、6.5 节、6.6 节,研究生教学适当增加 6.3.1 节、6.3.2 节、6.4.3 节。

本章难点在实际应用会涉及较多的综合知识,安全协议在这里已经不是一种单纯的研究内容,而是要将其实现。建议采用课程项目的方式讲授。

## 习题与思考题

1. 实际应用中的安全协议选择主要由什么因素决定?
2. 试述理论上安全的安全协议在实际应用中的安全性。
3. 举例说明一次性登录技术在安全协议方面的作用。
4. 试比较 SSL 与 SET 协议。
5. 试对 iKP 协议体系的安全性进行分析。
6. 通过对认证协议的理解,试说明非否认协议与认证协议的主要区别表现在哪些方面?

7. PKI 包含的基本内容有哪些？试述其作用。

8. 根据对数字现金安全需求的分析，试对本章所讲的相关协议之一设计一个例子并进行说明。

9. 试述 PKI 在安全电子支付协议中的作用。

10. 请列举 VPN 应用中应用的安全协议例子。

11. 试举例说明防火墙中所使用的安全协议。

# 附　　录

## A　AES 分组密码算法

分组密码，习惯上也称块密码，从 20 世纪 70 年代中期开始至今已有 30 余年的研究历史，其基本原理是将明文消息经编码表示后的二进制序列 $m_0, m_1, \cdots, m_i, \cdots$ 划分成若干固定长度的组（或块）$m=(m_0, m_1, \cdots, m_{l-1})$，各组分别在密钥 $k=(k_0, k_1, \cdots, k_{r-1})$ 的控制下转换成长度为 $n$ 的密文分组 $c=(c_0, c_1, \cdots, c_{n-1})$。其本质是一个从明文空间（$l$ 长的比特串的集合）$M$ 到密文空间（$n$ 长的比特串的集合）$C$ 的映射，该映射由密钥确定。分组密码简化框图如图 A-1 所示。

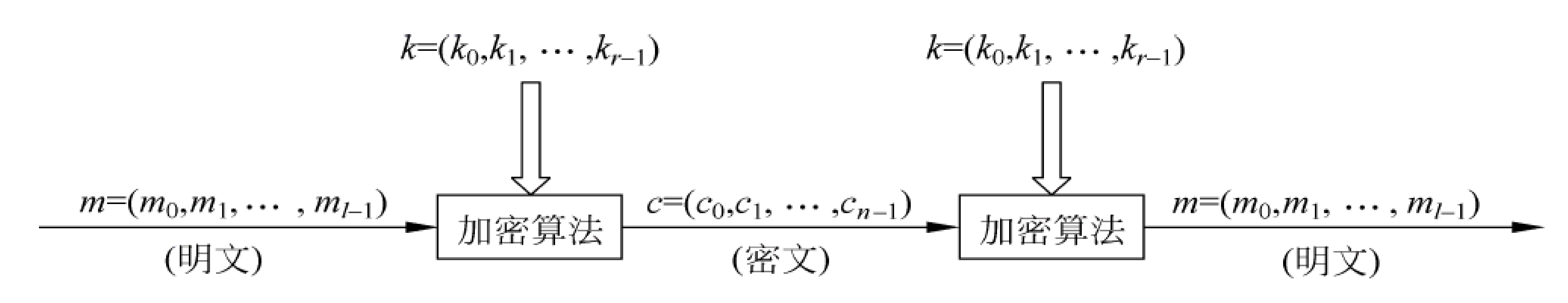



图 A-1　分组加密简化框图

目前，常用的分组密码有 DES[RFC 1829、RFC 1969、RFC 2419、RFC 3217]、AES[FIPS-197、RFC 3268、RFC 3602、RFC 3394]、IDEA[RFC 3058]等。

下面就具体介绍一下高级加密标准 AES(advanced encryption standard)。

1997 年 4 月，美国国家标准和技术研究所（NIST）开始征集先进加密标准（AES）算法来替代已不安全的 DES 算法。1998 年 5 月，NIST 宣布接收 15 个新的候选算法并提请全世界密码研究界协助分析这些候选算法，包括对算法的安全性和效率特性进行初步检验。之后，NIST 考察了这些初步的研究成果，并选定 MARS，RC6，Rijndael，Serpent 和 Twofish 这 5 个算法作为参加决赛的算法。经公众对决赛算法进行更进一步的分析评论后，NIST 于 2000 年 10 月宣布 Rijndael 作为高级加密标准（AES）。

Rijndael 算法是一个数据块长度和密钥长度可变的分组迭代加密算法。数据块的长度和密钥的长度可分别设定为 128 位、192 位和 256 位。

Rijndael 算法在整体结构上采用的是代替置换网络构成圈函数，多圈迭代，如图 A-2 所示。每一圈由以下 3 层组成。

① 非线性层：进行 S 盒变换 ByteSub，由 16 个 S 盒并置而成，起到混淆的作用。

② 线性混合层：进行行移位变换（shift row）和列混合变换（mix column），以确保多圈之上的高度扩散。

③ 密钥加层：进行圈密钥加变换（add round key），将圈密钥简单地异或到中间状态上。

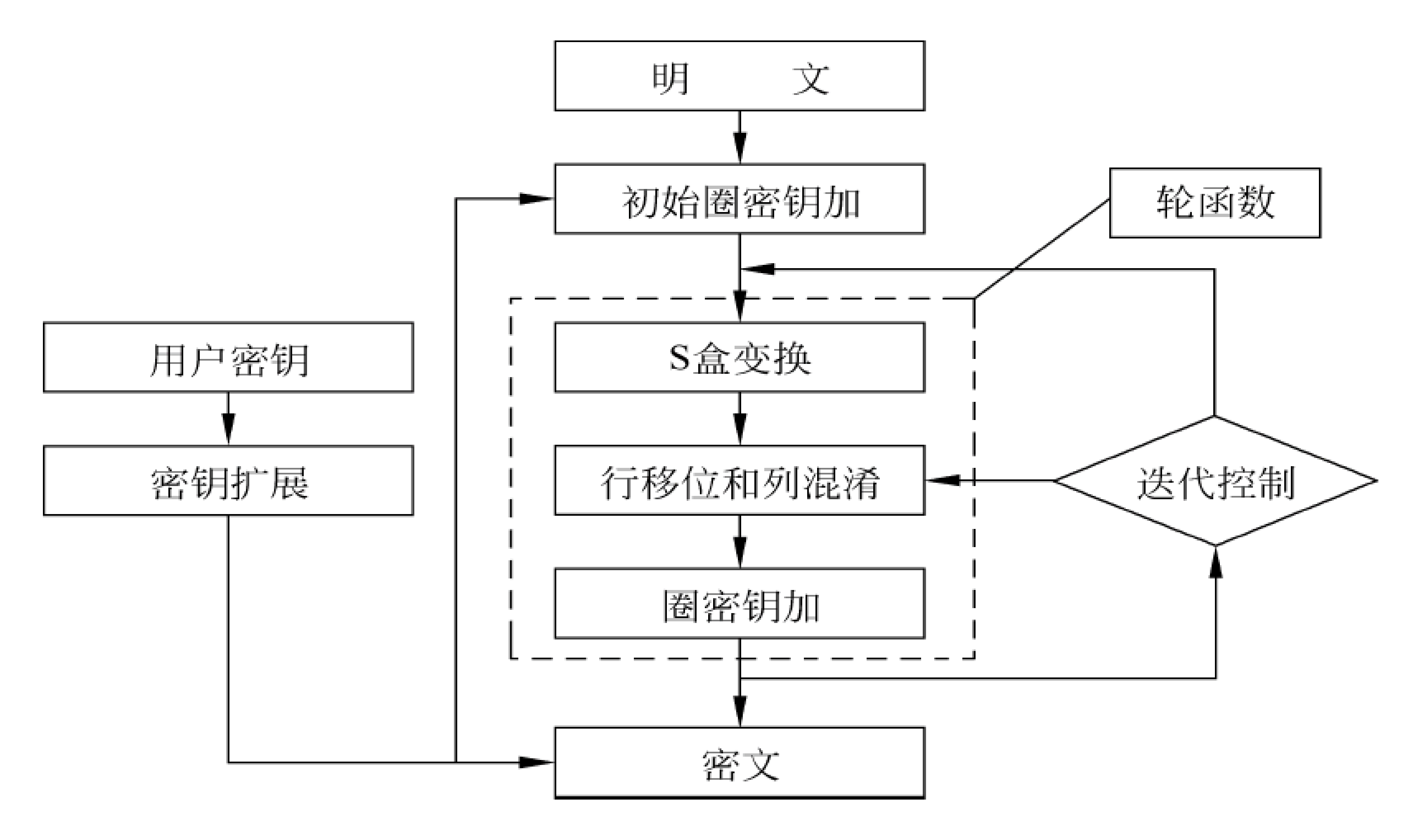



图 A-2　Rijndael 算法结构

## A.1　状态、密钥和轮数

Rijndael 的加密解密要经过多次数据变换操作，每一次变换操作都会产生一个中间结果，称这个中间结果为状态。算法的所有操作都以状态为对象进行。以 8 位二进制数为一个字节，4 个字节称为一个字，可以把一个状态表示为一个二维字节数组。该数组有 4 行、$N_b$ 列，每一列为一个字，列数由数据块长度除以 32 来决定。数据块长度分别为 128、192、256 位时的状态如表 A-1、表 A-2 和表 A-3 所示，$N_b$ 分别为 4、6 和 8。可以看出，不同情况下数组都是 4 行。

**表 A-1　数据块长度为 128 时的状态**

| | | | |
|---|---|---|---|
| $a_{00}$ | $a_{01}$ | $a_{02}$ | $a_{03}$ |
| $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{13}$ |
| $a_{20}$ | $a_{21}$ | $a_{22}$ | $a_{23}$ |
| $a_{30}$ | $a_{31}$ | $a_{32}$ | $a_{33}$ |

**表 A-2　数据块长度为 192 时的状态**

| | | | | | |
|---|---|---|---|---|---|
| $a_{00}$ | $a_{01}$ | $a_{02}$ | $a_{03}$ | $a_{04}$ | $a_{05}$ |
| $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{13}$ | $a_{14}$ | $a_{15}$ |
| $a_{20}$ | $a_{21}$ | $a_{22}$ | $a_{23}$ | $a_{24}$ | $a_{25}$ |
| $a_{30}$ | $a_{31}$ | $a_{32}$ | $a_{33}$ | $a_{34}$ | $a_{35}$ |

**表 A-3　数据块长度为 256 时的状态**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $a_{00}$ | $a_{01}$ | $a_{02}$ | $a_{03}$ | $a_{04}$ | $a_{05}$ | $a_{06}$ | $a_{07}$ |
| $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{13}$ | $a_{14}$ | $a_{15}$ | $a_{16}$ | $a_{17}$ |
| $a_{20}$ | $a_{21}$ | $a_{22}$ | $a_{23}$ | $a_{24}$ | $a_{25}$ | $a_{26}$ | $a_{27}$ |
| $a_{30}$ | $a_{31}$ | $a_{32}$ | $a_{33}$ | $a_{34}$ | $a_{35}$ | $a_{36}$ | $a_{37}$ |

数据块按顺序 $a_{00}, a_{10}, a_{20}, a_{30}, a_{01}, a_{11}, a_{21}, a_{31}, a_{02}, \cdots$ 的顺序映射为状态中的字节。在加密结束时，密文也按照同样的顺序从状态中取出。

类似地，密钥也可以表示为 4 行，$N_k$ 列的二维字节数组，其中，$N_k$ 等于密钥块长度除以 32。密钥块长度分别为 128，192，256 位时的字节数组如表 A-4、A-5 和 A-6 所示，$N_k$ 分别为 4、6 和 8。

**表 A-4　密钥块长度为 128 时的状态**

| | | | |
|---|---|---|---|
| $k_{00}$ | $k_{01}$ | $k_{02}$ | $k_{03}$ |
| $k_{10}$ | $k_{11}$ | $k_{12}$ | $k_{13}$ |
| $k_{20}$ | $k_{21}$ | $k_{22}$ | $k_{23}$ |
| $k_{30}$ | $k_{31}$ | $k_{32}$ | $k_{33}$ |

**表 A-5　密钥块长度为 192 时的状态**

| | | | | | |
|---|---|---|---|---|---|
| $k_{00}$ | $k_{01}$ | $k_{02}$ | $k_{03}$ | $k_{04}$ | $k_{05}$ |
| $k_{10}$ | $k_{11}$ | $k_{12}$ | $k_{13}$ | $k_{14}$ | $k_{15}$ |
| $k_{20}$ | $k_{21}$ | $k_{22}$ | $k_{23}$ | $k_{24}$ | $k_{25}$ |
| $k_{30}$ | $k_{31}$ | $k_{32}$ | $k_{33}$ | $k_{34}$ | $k_{35}$ |

**表 A-6　密钥块长度为 256 时的状态**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $k_{00}$ | $k_{01}$ | $k_{02}$ | $k_{03}$ | $k_{04}$ | $k_{05}$ | $k_{06}$ | $k_{07}$ |
| $k_{10}$ | $k_{11}$ | $k_{12}$ | $k_{13}$ | $k_{14}$ | $k_{15}$ | $k_{16}$ | $k_{17}$ |
| $k_{20}$ | $k_{21}$ | $k_{22}$ | $k_{23}$ | $k_{24}$ | $k_{25}$ | $k_{26}$ | $k_{27}$ |
| $k_{30}$ | $k_{31}$ | $k_{32}$ | $k_{33}$ | $k_{34}$ | $k_{35}$ | $k_{36}$ | $k_{37}$ |

算法中的加密解密变换是迭代进行的，它的重复次数称为圈数，用 $N_r$ 表示。$N_r$ 由 $N_b$ 和 $N_k$ 共同决定，其具体的数值如表 A-7 所示。

**表 A-7　算法变换的圈数**

| $N_r$ | $N_b=4$ | $N_b=6$ | $N_b=8$ |
|---|---|---|---|
| $N_k=4$ | 10 | 12 | 14 |
| $N_k=6$ | 12 | 12 | 14 |
| $N_k=8$ | 14 | 14 | 14 |

## A.2　圈变换

Rijndael 加密算法中的圈变换由 4 个不同的变换所组成，分别为 Byte Sub，Shift Row，Mix Column 和 Add Round Key。用伪 C 语言可将它表示为：

```
Round(State,RoundKey)
    { ByteSub(State);
      ShiftRow(State);
      MixColumn(State);
```

```
        AddRoundKey(State,RoundKey);}
```

加密算法的最后一轮略有不同,去掉了 Max Column(State),具体如下所示:

```
FinalRound(State,RoundKey)
  {ByteSub(State);
      ShiftRow(State);
      AddRoundKey(State,RoundKey);}
```

下面分别对每个变换的具体算法进行介绍。

## A.3 字节代换

字节代换(Byte Sub)独立地对状态的每个字节即状态数组中的每个元素进行非线性变换,又称为 S 盒变换。它分为两步进行。

(1) 首先,将字节看作 $GF(2^8)$上的元素,映射到自己的乘法逆元,"00"映射到自己。$GF(2^8)$上的二进制多项式 $b(x)$的乘法逆(记为 $b^{-1}(x)$)为满足 $a(x)b(x) \bmod m(x)=1$ 的多项式 $a(x)$。

(2) 其次,对字节做如下的仿射变换:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

图 A-3 是 $N_b$ 为 6 时字节代换的示意图。

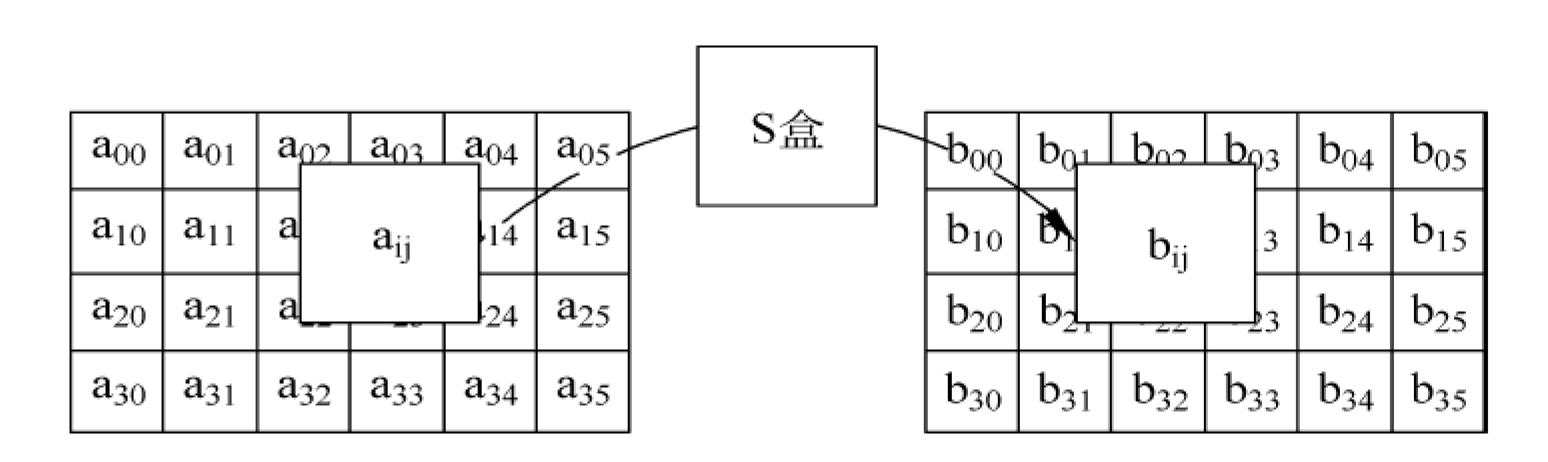



图 A-3 字节代换示意图($N_b=6$)

## A.4 行移位

行移位(Shift Row)将状态数组的每一行进行移位,移位量随不同的行而不同。第 0 行不移动,第 1 行循环左移 $C_1$ 个字节,第 2 行循环左移 $C_2$ 个字节,第 3 行循环左移 $C_3$ 个字节。位移量 $C_1$、$C_2$、$C_3$ 的取值与 $N_b$ 有关,它们的关系如表 A-8 所示,图 A-4 是 $N_b$ 为 6 时行移位的示意图。

表 A-8　对应于不同分组长度的位移量

| $N_b$ | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|
| 4 | 1 | 2 | 3 |
| 6 | 1 | 2 | 3 |
| 8 | 1 | 3 | 4 |

图 A-4　行移位示意图（$N_b=6$）

## A.5　列混合

在列混合（Mix Column）变换中，将状态阵列的每个列视为 $GF(2^8)$ 上的多项式，再与一个固定的多项式 $c(x)$ 进行模 $x^4+1$ 乘法。Rijndael 的设计者给出的 $c(x)$ 为（系数用十六进制表示）：

$$c(x)='03'x^3+'01'x^2+'01'x+'02'$$

因为 $c(x)$ 与 $x^4+1$ 是互素的，从而保证 $c(x)$ 存在逆多项式 $d(x)$，而 $c(x)d(x)=1 \bmod x^4+1$。列混合运算也可写为矩阵乘法。设 $b(x)=c(x)\otimes a(x)$，则

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}=\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

图 A-5 是 $N_b$ 为 6 时的列混合的示意图。

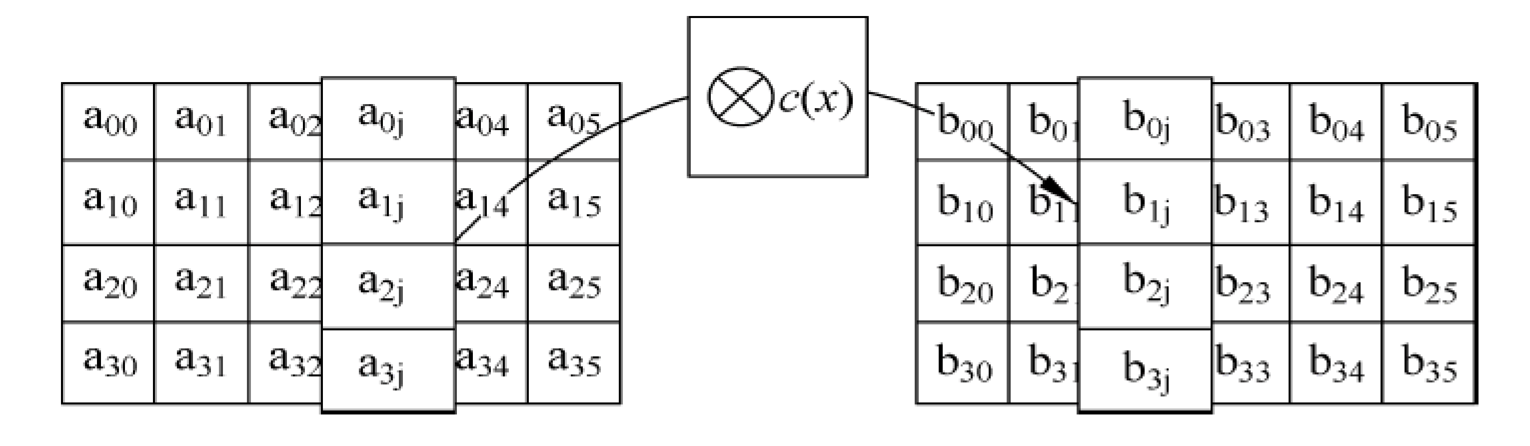



图 A-5　列混合示意图（$N_b=6$）

## A.6　密钥加

密钥加（Add Round Key）是将圈密钥与状态进行逐比特异或的过程。圈密钥根据密钥产生算法通过密钥得到。圈密钥的长度与数据块长度相等。图 A-6 所示是密钥加运算的示意图。

| | | | | | |
|---|---|---|---|---|---|
| $a_{00}$ | $a_{01}$ | $a_{02}$ | $a_{03}$ | $a_{04}$ | $a_{05}$ |
| $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{13}$ | $a_{14}$ | $a_{15}$ |
| $a_{20}$ | $a_{21}$ | $a_{22}$ | $a_{23}$ | $a_{24}$ | $a_{25}$ |
| $a_{30}$ | $a_{31}$ | $a_{32}$ | $a_{33}$ | $a_{34}$ | $a_{35}$ |

$\oplus$

| | | | | | |
|---|---|---|---|---|---|
| $k_{00}$ | $k_{01}$ | $k_{02}$ | $k_{03}$ | $k_{04}$ | $k_{05}$ |
| $k_{10}$ | $k_{11}$ | $k_{12}$ | $k_{13}$ | $k_{14}$ | $k_{15}$ |
| $k_{20}$ | $k_{21}$ | $k_{22}$ | $k_{23}$ | $k_{24}$ | $k_{25}$ |
| $k_{30}$ | $k_{31}$ | $k_{32}$ | $k_{33}$ | $k_{34}$ | $k_{35}$ |

=

| | | | | | |
|---|---|---|---|---|---|
| $b_{00}$ | $b_{01}$ | $b_{02}$ | $b_{03}$ | $b_{04}$ | $b_{05}$ |
| $b_{10}$ | $b_{11}$ | $b_{12}$ | $b_{13}$ | $b_{14}$ | $b_{15}$ |
| $b_{20}$ | $b_{21}$ | $b_{22}$ | $b_{23}$ | $b_{24}$ | $b_{25}$ |
| $b_{30}$ | $b_{31}$ | $b_{32}$ | $b_{33}$ | $b_{34}$ | $b_{35}$ |

图 A-6 密钥加运算示意图($N_b=6$)

## A.7 圈密钥产生算法

圈密钥产生是指从种子密钥得到圈密钥的过程。它分两步进行：密钥扩展和圈密钥选择，且遵循以下原则：

(1) 由于要进行 $N_r+1$ 次密钥加变换，每次所需的密钥长度等于数据块长度，因此圈密钥的比特总长度为数据块长度与 $N_r+1$ 的乘积。例如，对于 128 位的分组长度，圈密钥的总长度为 1408 位。

(2) 首先将密钥扩展为一个扩展密钥。

(3) 再从扩展密钥中选出圈密钥：第一个圈密钥由前面的 $N_b$ 个字组成，第二个圈密钥由接下来的 $N_b$ 个字组成，依此类推。

## A.8 密钥扩展

扩展密钥是以 4 字节字为元素的一维阵列，表示为 $W[N_b\times(N_r+1)]$。其中，前 $N_k$ 个字为种子密钥，后面的每个字都由它前面的字经过递归方式定义。根据 $N_k\leqslant6$ 和 $N_k>6$ 两种情况，扩展算法略有不同。

(1) 当 $N_k\leqslant6$ 时，扩展算法如下：

```
KeyExpansion(byte Key[4 * Nk],W[Nb * (Nr + 1)])
{
     for(i = 0;i<Nk;i ++ )
        W[i] = (Key[4 * i],Key[4 * i + 1],Key[4 * i + 2],Key[4 * i + 3]);
     for(i = Nk;i<Nb * (Nr + 1);i ++ )
     {
       temp = W[i - 1];
       if(i % Nk == 0)
            temp = SubByte(RotByte(temp))^Rcon[i/Nk];
       W[i] = W[i - Nk]^temp;
     }
}
```

其中，$Key[4*N_k]$为种子密钥，看作以字节为元素的一维阵列。函数 SubByte()返回 4 字节字，其中每一个字节都是用 Rijndael 的 S 盒作用到输入字对应的字节得到。函数 RotByte()也返回 4 字节字，该字由输入的字循环移位得到，即当输入字为(a,b,c,d)时，输出字为(b,c,d,a)。

从算法可以看到，扩展密钥的前 $N_k$ 个字即为种子密钥，之后的每个字 W[i]等于前

一个字 W[i－1]与 $N_k$ 个位置之前的字 W[i－$N_k$]的异或；不过当 i/$N_k$ 为整数时，需先将前一个字 W[i－1]经过以下一系列的变换。

1 字节的循环移位 RotByte→用 S 盒进行变换 SubByte→异或轮常数 Rcon[i/$N_k$]。

(2) 当 $N_k$>6 时，扩展算法如下：

```
KeyExpansion(byte Key[4 * Nk],W[Nb * (Nr + 1)])
{
    for(i = 0;i<Nk;i ++ )
        W[i] = (Key[4 * i],Key[4 * i + 1],Key[4 * i + 2],Key[4 * i + 3]);
    for(i = Nk;i<Nb * (Nr + 1);i ++ )
    {
        temp = W[i - 1];
        if(i % Nk == 0)
           temp = SubByte(RotByte(temp))^Rcon[i/Nk];
        else if(i % Nk == 4)
           temp = SubByte(temp);
        W[i] = W[i - Nk]^temp;
    }
}
```

$N_k$>6 相比较于 $N_k$≤6 的情况，其密钥扩展算法的区别在于：当 i 为 $N_k$ 的 4 倍数时，须先将前一个字 W[i－1]经过 SubByte 变换。

以上两个算法中，Rcon[i/$N_k$]为轮常数，它的值与 $N_k$ 无关，定义为(字节用十六进制表示，同时理解为 GF($2^8$)上的元素)

$$\text{Rcon}[i] = (\text{RC}[i], \text{'00'}, \text{'00'}, \text{'00'})$$

其中，RC[i]时 GF($2^8$)中值为 $x^{i-1}$ 的元素，因此

$$\text{RC}[1] = 1(\text{即'01'})$$

$$\text{RC}[i] = x\ (\text{即'02'})\ \text{RC}[i-1] = x^{i-1}$$

## A.9　圈密钥的选取

圈密钥 i 由圈密钥缓冲区 W[$N_b$ * i]到 W[$N_b$ * (i＋1)－1]的字组成。例如，$N_b$＝6 且 $N_k$＝4 的圈密钥选择如图 A-7 所示。

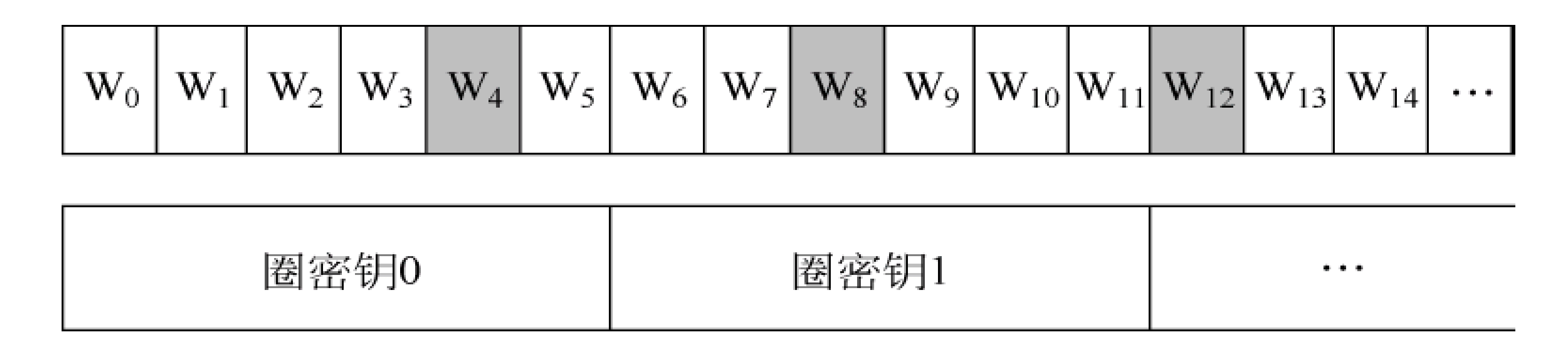



图 A-7　$N_b$＝6 且 $N_k$＝4 时的密钥扩展与圈密钥选取

## A.10　Rijndael 加密算法

根据前面的描述，可以将 Rijndael 的加密算法概括成以下组成部分：

(1) 一个初始圈密钥加法变换。

（2）$N_{r-1}$圈的圈变换。

（3）最后一圈变换。

用伪C语言表示为：

```
Rijndael(State,CipherKey)
{
    KeyExpansion(CipherKey,ExapndKey)
    AddRoundKey(State,ExpandKey)
    for(i = 1;i<Nr;i ++ )
        Round(State,ExpandKey[Nb * i])
        {
            ByteSub(State);
            ShiftRow(State);
            MixColumn(State);
            AddRoundKey(State,ExpandKey[Nb * i]);
        }
    FinalRound(State,ExpandKey[Nb * Nr])
    {
        ByteSub(State);
        ShiftRow(State);
        AddRoundKey(State,ExpandKey[Nb * Nr]);
    }
}
```

## A.11 Rijndael 解密算法

Rijndael解密算法的结构与加密算法的结构相同，其中的变换为加密算法变换的逆变换，且密钥扩展策略稍有不同。

### 1. 逆变换

（1）字节代换（Byte Sub）的逆是Rijndael的S盒逆作用到状态的每个字节上。首先，进行下式中的逆变换

$$\left\{\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}\right\} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}$$

然后，再取$GF(2^8)$上的乘法逆。上式中的矩阵就是前面字节代换小节（A.3）中矩阵的逆矩阵。

（2）行移位（Shift Row）的逆是状态的后3行分别移动$N_b-C_1$，$N_b-C_2$和$N_b-C_3$

个字节。

(3) 列混合(Mix Column)的逆类似于 Mix Column 自己，状态的每列都乘以一个固定的多项式 $d(x)$：

$$d(x) = '0B'x^3 + '0D'x^2 + '09'x + '0E'$$

可以验证，$d(x)$与前面列混合小节(A.5)中的 $c(x)$的积等于单位元‘01’。所以，$d(x)$是 $c(x)$的逆多项式。

(4) 密钥加(Add Round Key)的逆就是它自己。

### 2. 逆圈变换的定义

由上面的结论，逆圈变换的定义如下：

```
Inv_Round(State,Inv_RoundKey)
{
        InvByteSub(State);
        InvShiftRow(State);
        InvMixColumn(State);
        AddRoundKey(State,Inv_RoundKey);
}
```

最后一圈的逆变换如下：

```
Inv_Round (State,Inv_RoundKey)
{
        InvByteSub(State);
        InvShiftRow(State);
        AddRoundKey(State,Inv_RoundKey);
}
```

### 3. 解密算法

由以上的描述，可将 Rijndael 的解密算法表述如下：

```
Inv_Rijndael(State,CipherKey)
{
     Inv KeyExpansion(CipherKey,Inv_ExpandKey);
     AddRoundKey(State,Inv_ExpandKey[Nb * Nr]);
     for(i = 1;i<Nr;i ++ )
        Inv_Round(State,Inv_ExpandKey[Nb * i]);
        {
             InvByteSub(State);
             InvShiftRow(State);
             InvMixColumn(State);
             AddRoundKey(State,Inv_ExpandKey[Nb * i]);
        }
        Inv_FinalRound(State,Inv_RoundKey)
             {
                  InvByteSub(State);
```

```
            InvShiftRow(State);
            AddRoundKey(State,Inv_ExpandKey);
        }
    }
```

其中,解密算法的密钥扩展定义为:

(1) 加密算法的密钥扩展。

(2) 把 Inv Mix Column 应用到除第一和最后一圈外的所有圈密钥上。

用伪 C 语言表示如下:

```
Inv_KeyExpansion(CipherKey,Inv_ExpandKey)
    {
        Key_Expansion(CipherKey,Inv_ExpandKey);
        for(i = 1;i<Nr;i ++ )
        InvMixColumn(Inv_ExpandKey + Nb * i);
    }
```

# B 公钥密码——椭圆曲线加密算法

公开密钥密码,简称公钥密码,该体制在加密和解密时使用不同密钥,即加密功能和解密功能分开。其中,每个用户保存一对密钥,即公钥 PK 和私钥 SK,PK 是公开信息,不需要保密,且通过给定的公钥要确定出私钥在计算上是不可行的。

公钥密码体制有两种基本的模型:一种是加密模型,一种是认证模型,分别如图 B-1 和图 B-2 所示。

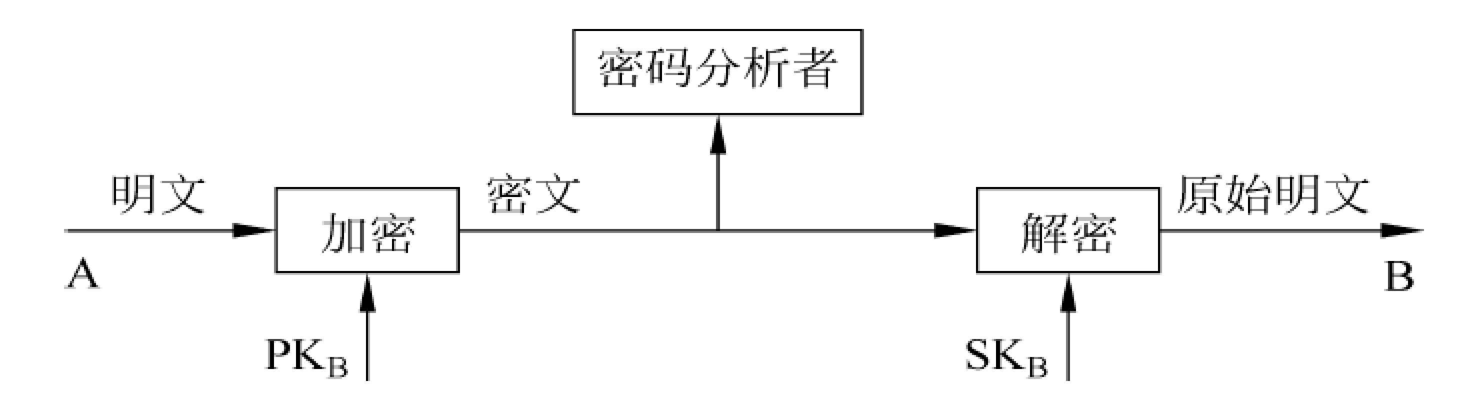



图 B-1 加密模型

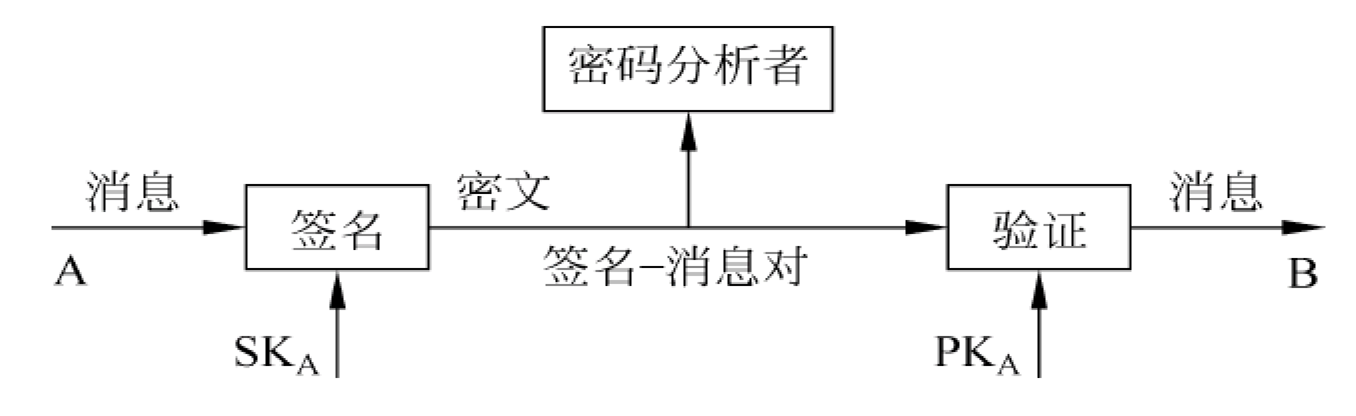



图 B-2 认证模型

目前,比较常用的公钥密码算法包括 RSA[RFC 78]、ElGamal[RFC 2440]、Diffie-Hellman[RFC 2631、RFC 2875]以及 ECC[RFC 3278]等。下面将主要介绍 ECC 体制的相关内容。

1985 年，N. Koblitz 和 V. Miller 分别独立提出了椭圆曲线密码体制（ECC），其安全基于椭圆曲线离散对数问题的难解性。椭圆曲线指的是由 Weierstrass 方程：

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{B-1}$$

所确定的平面曲线，其中，系数 $a_i(i=1,2,3,4,6)$ 定义在某个域上，可以是有理数域、实数域、复数域，也可以是有限域等。

## B.1　椭圆曲线的选取

要建立椭圆曲线密码体制，首要的问题是选取一个合适的背景有限域 K 及在 K 上选取一条合适的椭圆曲线 E/K。从实用观点看，K 有两种选择：大素域 $Z_p$ 或特征为 2 的有限域 GF($2^n$)。从近年来的实践结果看，大素域更为有效一些。椭圆曲线的选取则要考虑安全性、实用性等诸多因素。有些密码体制（如 ElGamal 签名体制、DSS 签名体制）需要知道 E 的阶＃E(K)或＃E(K)的一个大素因子。另一些体制（如 Diffie-Hellman 密钥交换协议 ElGamal 密钥体制）虽不需要知道 E(K)的阶，但为避免 Pohlig-Hellman 攻击，需保证＃E(K)中有大素因子。椭圆曲线的选取现有两种可以考虑的方法。

### 1. 随机选取

随机选取一条椭圆曲线 E/K，计算其阶参数＃E(K)，直到获得满意的曲线为止。由于这种方法的随机性，从安全性角度来讲这是一种理想的方法。

Hasse 定理告诉我们一个关于＃E(K)的估计：令＃E(K)$=q+1-t$，则 $|t|\leqslant 2\sqrt{q}$。但要具体求出＃E(K)却并非易事，Schoof 在这方面作出了开创性的成果。令 $\varphi$ 是 $E(\overline{F_q})$ 2(瓦)上的 Frobenjus 自同态：

$$\varphi:(x,y)\mapsto(x^q,y^q)\ \forall(x,y)\in E(\overline{F_q}) \tag{B-2}$$

其特征方程为 $\varphi^2-t\varphi+q=0$，其中 $t\in Z$，$|t|\leqslant 2\sqrt{q}$，$\#E(F_q)=q+l-t$。设 $l$ 是一个小素数，$E(l)$为 $E(\overline{F_q})$的 $l$—扭点构成的子群，通过将 $\varphi$ 限制在 $E(l)$中。利用搜索可求出 $t^l$ 满足 $t^l\equiv t \bmod l$，即对于小素数 $l$ 可求出 $t \bmod l$。

Schoof 算法的基本思想就是对一系列小素数 $l=3,5,7,\cdots,L$，其中，$L$ 满足 $\prod\limits_{\substack{l\leqslant L\\ l\neq p}} l \geqslant 4\sqrt{q}$，求出 $t \bmod l$，从而由中国剩余定理得到＃E(K)。

Schoof 的这个算法具有时间复杂度 $O(\log_2 q)$，理论上是个有效算法，在实际中却不实用，但这个方法指出了求算＃E(K)的一个努力方向，引起了极大关注。自此以后，围绕计算 $t \bmod l$ 已有大量成果发表出来，并且在方法的实现上有了较大进展。

### 2. 构造给定阶的椭圆曲线

Atkin 和 Morain 的论文“Elliptic Curves and Prrmality Proving”，使人们看到了获得密码体制所需要的椭圆曲线的另一条途径，该文提出的利用复乘构造素域 $Z_p$ 上具有特定阶椭圆曲线的思想及方法已引起了广泛关注，并被多篇论文讨论改进。密码标准 IEEE P1363 也采用了该策略作为生成椭圆曲线的方法之一。椭圆曲线的构造方法如下：

设 D 是一个负奇基本判别式，$H_p(X)$表示 D 的 Hilbert 类多项式，又设 $p$ 是一个素数，若整数 x，y 满足 $4p=x^2+Dy^2$，则对只 $H_p(X)$的任意关于模 $p$ 的根 $j$ 必存在 $j$ 一，不变量为 $j$ 的椭圆曲线 $Z_p$ 满足

$$4\#E(Z_p) = (x-2)^2 + Dy^2 \tag{B-3}$$

虽然椭圆曲线不能由 $j$ 一不变量唯一确定，通过 $j$ 一不变量 $j$ 找出满足式(B-3)的椭圆曲线是容易的。事实上 $j$ 一不变量为 $j$ 的椭圆曲线恰构成 2 个等势的同构类。

下面是构造素域 $Z_p(p>3)$上的素数阶椭圆曲线的方法的简化描述。

(1) 取定负奇基本判别式 D，使其具有小的类数(比如，D =19)。

(2) 在适当范围内，随机选取整数 $x,y$ 令 $4p=x^2+Dy^2$，检测 $q$ 的素性，直到 $q$ 是素数为止。

(3) 令 $4p=(x+2)^2+Dy^2$，检测 $p$ 的素性，若 $p$ 不是素数，返回(2)，直到 $p$ 为素数。

(4) 计算 $H_p(X)\equiv 0 \bmod p$ 的根 $j$ 由于 D 具有小的类数，该方程易解。

(5) 构造 $j$ 一不变量为 $j$ 的椭圆曲线 $E$：$y^2=x^3+ax+b$，取随机数 $c\ni Z_p^*$。在 $E'$：$y^2=x^3+c^2ax+c^3b$ 上任取一点 $P\neq 0$，判断 $qP=0$，直到成立。

可以证明，算法结束时 $E'$即为 $Z_p$ 上 $q$ 阶椭圆曲线。

该算法具有较高的效率，可轻易在大素域 $Z_p$ 上构造出素数阶椭圆曲线。

受到 MOV 归约的启示，人们对利用复乘构造椭圆曲线的方法存在着某些疑虑。对 $p$ 的形状的限制，对 D 的限制是否会影响体制的安全性？国内外密码学家对此广泛关注。但到目前为止，没有任何线索说明这种曲线存在弱点。

这里介绍的椭圆曲线密码是基于有限域 $F_p$ 上椭圆曲线有理点群的一种密码系统。下面给出有限域 $F_p$ 上的椭圆曲线 $E$。为简单起见，设 $p$ 是一个大于 3(即 $p\geqslant 5$)的素数，有限域 $F_p$ 上的椭圆曲线 $E$ 是定义在仿射平面上的 3 次方程 $E$：

$$y^2 = x^3 + ax + b \tag{B-4}$$

所有解与无穷远点 $o$ 构成的点集合，记作

$$E(F_p) = \{(x,y) \mid y^2 = x^3 + ax + b, (x,y) \in F_p * F_p\} \cup \{o\} \tag{B-5}$$

这是一个有限集合，其中，$p$ 是素数，$a,b\in F_p$ 且满足 $4a^3+27b^2\neq 0\in F_p$。$E(F_p)$的点数用$\#E(F_p)$表示。由 Hasse 定理可知

$$P+1-2\sqrt{P} \leqslant \#E(F_p) \leqslant P+1+2\sqrt{P} \tag{B-6}$$

点集合 $E(F_p)$对应下面的加法规则构成一个群，即

(1) 单位元 $o$：$o+o=o$，对任一 $P\in E(F_p)$，有 $o+P=P+o=P$。

(2) 对任一 $P\in E(F_p)$，$P=(x,y)\neq o$，则存在$-P\in E(F_p)$，$-P=(x,-y)$满足 $P+(-P)=o$(即点$(x,y)$逆为$(x,-y)$)。

(3) 设 $P,Q,R\in E(F_p)$，则$(P+Q)+R=P+(Q+R)$。

(4) (两个不同且不互逆的点的加法规则)令 $P,Q\in E(F_p)$，$P=(x_1,y_1)$，$Q=(x_2,y_2)$，且满足 $x_1\neq x_2$，则 $R=P+Q=(x_3,y_3)\in E(F_p)$，其中

$$\lambda = (y_2 - y_1)/(x_2 - x_1) \tag{B-7}$$

$$x_3 = \lambda^2 - x_1 - x_2 \tag{B-8}$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \tag{B-9}$$

（5）（倍点规则）令 $P=(x_1,y_1)\in E(F_p)$，$y_1\neq 0$ 则 $R=2P=(x_3,y_3)\in E(F_p)$，其中

$$\lambda = (3x_1^2 + a)/2y_1 \tag{B-10}$$

$$x_3 = \lambda^2 - 2x_1 \tag{B-11}$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \tag{B-12}$$

可通过椭圆曲线加法群的几何意义验证群 $E(F_p)$ 是阿贝尔群，即对任意 $P,Q\in E(F_p)$，有 $P+Q=Q+P$。如果 $\sharp E(F_p)=p+1$，曲线就称为是超奇异的，否则称为是非超奇异的。文中以下讨论的都是非奇异的满足上述条件的椭圆曲线。

在 $E(F_p)$ 中选一个点 $P$，称为基点，记 $P$ 的阶为 $n$，通常要求 $n$ 是一个大素数。每个用户选取一个整数 $e(1\leqslant e<n)$ 作为其私钥，而以点 $D=eP$ 作为其公钥，这样就形成一个椭圆曲线公钥密码系统（ECC）。每个用户的公钥都是该系统的公开参数，每个用户的私钥都是保密的。

假设用户 $A$ 欲将明文 $m(0<m<p)$ 加密后发送给 $B$，$A$ 首先要查得 $B$ 的公钥 $D_B$，然后进行以下的加密运算：

（1）取随机数 $k\in Z$，计算 $kP=(x_1,y_1)$。

（2）计算 $kD_B=(x_2,y_2)$。

（3）计算密文 $c=m\oplus x_2$（将 $m$ 和 $x_2$ 用二进制表示，然后按位模 2 加），将 $(c,x_1,y_1)$ 发送给 $B$。

$B$ 收到 $A$ 发来的信息后，进行下述的运算：

（1）计算 $e_B(x_1,y_1)=(x_2,y_2)$，$e_B$ 为 $B$ 的私钥。

（2）计算 $c=m\oplus x_2$，得到明文 $m$。

因为 $e_B(x_1,y_1)=e_BkP=kD_B=(x_2,y_2)$，则上述加、解密运算都是正确的。

假设点 $Q$ 是 $E(F_p)$ 上为 $P$ 的倍数的点，即存在整数 $k(0\leqslant k\leqslant n-1)$，使得 $Q=kP\in E(F_p)$。由 $k$ 和 $P$ 计算 $Q=kP$ 可以利用加法公式，而由 $Q$ 和 $P$ 来求 $k$ 是一个困难的问题（当 $n$ 很大时），这就是椭圆曲线上的离散对数问题。从椭圆曲线密码系统的形成很容易了解到，ECC 的安全性是建立在离散对数计算难度之上，如果离散对数可以计算，从一个用户的公钥就可得到他的私钥，ECC 就不安全了。在应用椭圆曲线公钥密码时，最主要的计算量在计算 $kP$。

椭圆曲线密码体制的另一个重要用途是进行数字签名。在计算机网络通信中，数字签名可用于确认发信人的身份；发现在传输过程中，信息 $m$ 是否被非法篡改；具有不可抵赖、不可更改性。

椭圆曲线密码系统（ECC）相对于其他公钥密码系统，具有计算速度快，存储空间小，带宽要求低等优点，受到了人们的广泛关注，成为最有希望的公钥密码系统。与此同时存在很多因素影响椭圆曲线公钥密码系统的运算速度，如椭圆曲线系统的参数选取（所在的有限域，域元素的表示形式，椭圆曲线的形式）等。此外，还受所需要的安全强度，应用平台（软件，硬件或软硬结合），特殊的通信环境（如带宽）的限制。关于椭圆曲线公钥密码系统的研究主要有以下 3 个方向：椭圆曲线公钥密码体制的构造、椭圆曲线密码体制的分析和椭圆曲线公钥密码体制的快速实现。

## B.2 典型的椭圆曲线加密体制

现有的椭圆曲线密码体制均是从其他群中平移而来，并未针对 $E(K)$ 产生新型密码体制，而这种平移主要是对基于离散对数问题的密码体制，虽然也有 RSA 体制的平移，但并无实用及理论价值，下面给出了一些基于离散对数体制的讨论。

设 $G$ 是一个有限群，$a, b \in G$。若存在正整数 $n$ 使得 $a^n = b$，则称为群 $G$ 中是 $b$ 的以 $a$ 为底的离散对数，记为 $n = \log_a b$，给定 $a, b \in G$，求 $n = \log_a b$ 称为 $G$ 中的离散对数问题，特别地，若 $P, Q \in E(If)$，求 $n$ 使得 $nP = Q$，称为椭圆曲线离散对数问题。

将 ElGamal 加密体制直接平移到椭圆曲线群上，得到的密码体制将需要首先把加密的明文转化为椭圆曲线上的点，而后才能进行加密，这在实用上较为麻烦，为避免这个麻烦，Menezes 和 Vanstone 对该体制作了一点轻微的修改。下面介绍的 EC-ElGamal 体制采用了这种改进形式。

### 1. EC-ElGamal 加密体制

(1) 选取有限域 $K$、椭圆曲线 $E/K$ 及基点 $P \in E(K)$（这些参数可由一组用户公用）。

(2) 选取随机数 $a$，计算 $Q = aP$。

(3) 公开 $K, E, P, Q$ 作为公钥，密藏 $a$ 作为私钥。

假设 Alice 已建立了上述体制，给 Alice 发送秘密消息 $M = (M_1, M_2) \in K \times K$ 需完成如下步骤：

① 随机选取正整数 $k$。

② 计算 $kP, kQ = (x, y)$，若 $x = 0$ 或 $y = 0$ 返回第①步，直到 $x \neq 0, y \neq 0$。

③ 发送 $C = (kP, M_1 x, M_2 y)$ 给 Alice。

收到密文 $C$ 后，Alice 计算 $a(kP) = (x, y)$，进而得到明文 $M = (M_1, M_2)$。

### 2. EC-DSS 签名体制

(1) 选取有限域 $K$、椭圆曲线 $E/K$ 及基点 $P \in E(K)$，设 $<P>$ 是由 $P$ 生成的 $q$ 阶循环子群，$q$ 是一个大素数。

(2) 选取随机数 $x$，$0 < x < q$ 计算 $Q = xP$。

(3) 选取单向 Hash 算法 $H: M \to Z$，其中，$M$ 是消息空间，$Z$ 是整数集。并选取双射

$$g: <p> \to \{0, 1, \cdots, q-1\}$$

(4) $K, E, P, Q, q, g$ 均为公开信息。作为签名验证公开钥，$x$ 作为签名密钥。设有消息 $m \in M$，对 $m$ 的签名过程如下。

① 随机选取整数 $k$，$0 < k < q$。

② 计算 $R = kP$。

③ 解同余方程 $H(m) = -xg(R) + ks \bmod q$。

对 $m$ 的签名为 $(R, s)$，签名验证方程为

$$(H(m)s^{-1} \bmod q)P + (g(R)s^{-1} \bmod q)Q = R \tag{B-13}$$

## B.3 常见的椭圆曲线协议简介

(1) 椭圆曲线加密体制(elliptic curve encryption system,ECES)

其一般步骤如下。

① 选取合适的有限域 $F_q$,$E$,建立一个消息空间 $P$ 到椭圆曲线群的可逆嵌入映射,将信息编码:$\pi: P \to E, \pi: m \mapsto p_m$。

② 若 $E_k$,$D_k$ 是以离散对数为基础的加解密算法,将此算法换成 ECDLP,得:

$$E_k: E \to E, E_k: m \mapsto E_k(p_m) \tag{B-14}$$

$$D_k: E \to E, E_k: m \mapsto D_k(p_m) \tag{B-15}$$

③ 反编码:$\pi^{-1}: P \to E, \pi^{-1}: m \mapsto p_m$。

目前,还没有有效的编码方法,可行的有概率编码法。

(2) 椭圆曲线密钥交换体制(elliptic curve Diffie-Hellman,ECDH)

密钥交换体制如下:

选择 $p \approx 2^{180}$,$E_p(a,b): y^2 = x^3 + ax + b$,基点 $G$ 的阶为大素数。设 $A$,$B$ 两用户分别产生各自的公私钥对$(Q_A, n_A)$,$(Q_B, n_B)$,则密钥交换体制产生双方的公共密钥 $k$ 如下:

① $Q_A = n_A G; Q_B = n_B G$。

② $k = n_A Q_B = n_B Q_A = n_A n_B G$。

该方法并不能证明私钥的所属权。

(3) ECELG-ElGamal 加密体制

设 $\pi: P \to E, m \mapsto p_m$,是明文到曲线的嵌入,$G \in E$ 为基点。

① $A$ 选择 $n_A$,$Q_A = n_A G$。

② $B$ 若向 $A$ 发信息 $m$,则选择随机数 $k$,加密 $m$,发送$(kG, P_M + kQ_A)$。

③$A$ 解密消息:$(P_m + kQ_A) - n_A(kG) = P_m + kn_A G - n_A kG = P_m$。

(4) ECMO-Massey-Qmura 加密体制

它是类同于 RSA 的椭圆曲线加密算法。设 $E(F_q)$,$N = \# E(F_q)$,$\pi: P \to E, m \mapsto p_m$。

$A$ 选取 $e_A$,私钥 $d_A$ 内满足 $e_A d_A \equiv 1 \bmod N$,$1 < \bmod N$,$1 < d_A < N$,则

$$C_m = e_A P_m, \quad P_m = d_A C_m = e_A d_A P_m \tag{B-16}$$

(5) ECIES(elliptic curve integrated encryption scheme)

椭圆曲线综合加密标准是一种以密钥交换为基础的对称加密,首先用接收者的固有密钥对进行 ECDH 密钥交换,然后产生的密钥用于对称加密。

新型密钥交换体制(ECMQV)

MQV 将取代 DH 作为新的密钥交换体制,通常通信双方必须用他们的私钥来产生共享密钥,通过该体制可以证明私钥的所有权。这可以保证密钥交换提供更强的认证和保证恶意的一方不能伪装为第三方。并且只有很少的通信量和不重复、角色对称而且不需要加密和时间戳。

ECMQV 过程如下:假设双方 Alice 拥有椭圆曲线密钥对$(A,a)$,同样 Bob$(B,b)$,$P$ 为基点,已经通过 CA 认证中心验证拥有的公钥通过可信方式交换,私钥的所属权也得以证明;Alice 产生临时会话密钥对$(X,x)$$x$ 随机产生,$X = xP$;Bob 产生会话密钥对$(Y,y)$

随机产生，$Y=yP$。

Alice 计算 $S_A=(x+\overline{\gamma}a) \bmod n$（称为固有签名）（$n$ 为域大小）。

Bob 同样方式计算 $S_B=(y+\overline{\gamma}b) \bmod n$。

双方共同计算共享密钥：$K=hS_A(Y+\overline{\gamma}B)=hS_B(x+\overline{\chi}A)$（$h$ 为公因子，$\overline{x}$ 或 $\overline{y}$ 代表点 $X$ 或 $Y$ 的第一个域元素的前 $L$ 比特，$L=(\log_2^n)+1/2$）。

安全性：可以保证第三方不能伪造双方的私钥所属权，因为第三方没有双方的私钥 $a$，$b$，所以不能够计算出 $K$ 值。如果双方通信后得到的 $K$ 值相等，则说明双方的身份确定，密钥对无误；否则双方拒绝通信，从而防止第三者假冒通信双方中的一方。

通过该方式，可以确认临时密钥对的可靠性。通信过程如图 B-3 所示。

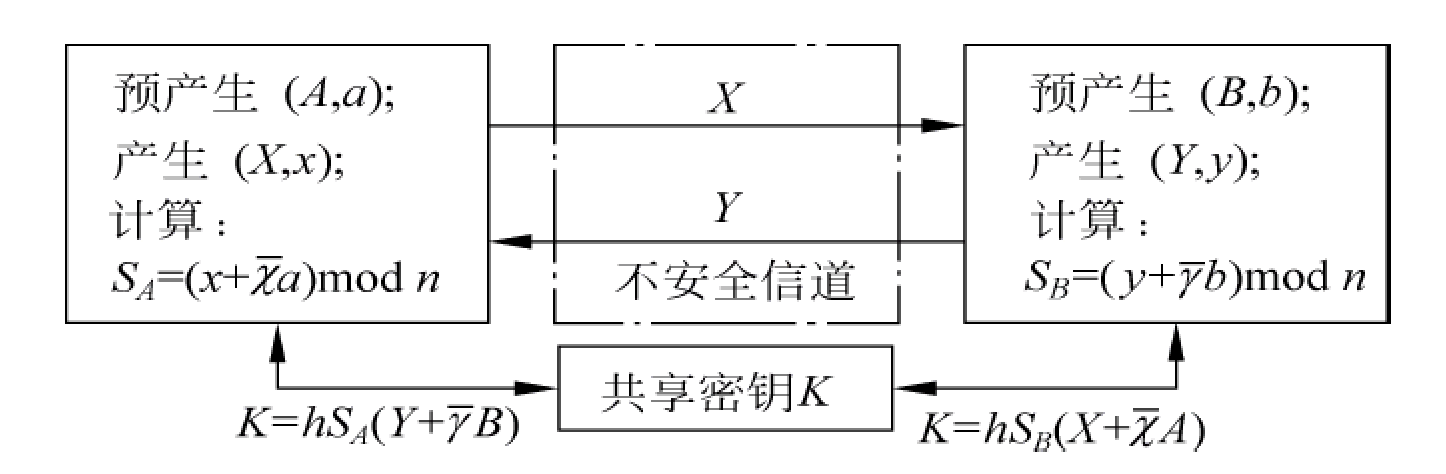



图 B-3　ECMQV 密钥交换体制

## B.4　椭圆曲线 Menezes-Vanstone 加密算法

Menezes-Vanstone 加密算法可以较好地发挥椭圆曲线加密的优势。它有两种类型。

(1) 素数域上非超奇椭圆曲线的 Menezes-Vanstone 加密算法

设消息 $(m_1,m_2)\in F_q\backslash\{0\}\times F_q\backslash\{0\}$，$P\in E(F_q)$ 是 $E$ 上一个 $n$ 阶点，关于 $P$ 的 DLP 问题是困难的。

用户 A，$B$ 选取各自的密钥 $l_A$、$l_B$，$l_A$、$l_B\in\{2,3,\cdots,n-2\}$；$n>2^{20}$。

用户 $A$ 计算 $Q_A=l_AP$，将 $Q_A$ 公开，用户 $B$ 计算 $Q_B=l_BP$，将 $Q_B$ 公开；用户的密钥对分别是：$(l_A,Q_A)$，$(l_B,Q_B)$。

$A$ 加密过程：

① $A$ 获取 $B$ 的公钥 $Q_B$，计算 $(x_1,y_1)=l_BQ_A$。

② 计算 $x_2=m_1x_1 \bmod q$，$y_2=m_2y_1 \bmod q$，且向 B 发送消息 $(Q_A,(x_2,y_2))$。

$B$ 解密过程：

① 计算 $(x_1,y_1)=l_AQ_B$。

② 计算 $m_1=x_2x_1^{-1} \bmod q$，$m_2=y_2y_1^{-1} \bmod q$，得到消息 $M=(m_1,m_2)$。

这一方案对任意 $(m_1,m_2)\in F_q\backslash\{0\}\times F_q\backslash\{0\}$ 都可以编码。

注意：

① 如果 $x_1y_1=0$，则 $A$ 重新选取公钥 $l_A$，但这种情形发生的概率很小。

② 这一方法的膨胀率是 2。

为弥补这两个缺陷，算法可改进为：用户 $B$ 向 $A$ 发送信息 $((Q_A)_x,l_{(y_1/x_1)},x_2,y_2)$ 而不是发送消息 $(Q_A,(x_2,y_2))$，其中，$(Q_A)_x$ 表示 $Q_A$ 的 $x$ 分量，$l_{(y_1/x_1)}$ 表示 $y_1$ 除以 $x_1$ 的某

个比特。这时,密文膨胀率为 1.5。

(2) $F_{2^m}$ 上非超奇椭圆曲线的 Menezes-Vanstone 密码算法

目前,常用的椭圆曲线 $E: y^2+xy=x^3+ax^2+b, b\in F_{2^m}^*, a\in\{0,r\}$,$r$ 的迹为 1。

通常,取 a=0,这样可以在倍点公式中简化计算。

加解密过程同上,但是在弥补缺陷发送($(Q_A)_x, l_{(y_1/x_1)}, x_2, y_2$)时,为了求取$(Q_A)_y$,选取域的正规基:

① 如果 $x=0$,则 $y^2=b$。记 $y=(y_0,y_1,\cdots,y_{m-1})$,$b=(b_0,b_1,\cdots,b_{m-1})$,则$(y_{m-1},y_0,y_1,\cdots,y_{m-2})=(b_0,b_1,\cdots,b_{m-1})$,得到唯一解:$y_0=b_1, y_1=b_2,\cdots,y_{m-2}=b_{m-1}, y_{m-1}=b_0$。

② 否则,对方程作变量代换$(x,y)\to(x,xz)$,得:$z^2+z=x+a+bx^{-2}$。令 $t=x+a+bx^{-2}$,解方程 $z^2+z=t$。

令 $z=(z_0,z_1,\cdots,z_{m-1})$,$t=(t_0,t_1,\cdots,t_{m-1})$,则:$(z_{m-1}+z_0, z_0+z_1,\cdots,z_{m-2}+z_{m-1})=(t_0,t_1,\cdots,t_{m-1})$。进一步可计算出 $y=zx$。

# C　部分 Hash 算法简介

现有的 Hash 算法主要有 MD 系列算法[RFC 1320、RFC 1321、RFC 1828],RIPEMD 算法[RFC 2286、RFC 2857],HAVAL 算法[FIPS-180],SHA 系列算法[RFC 3174、RFC 4509],Whirlpool 算法[RFC 3114、RFC 2634、FIPS-198],Tiger 算法[FIPS- 180]和 MDC 系列算法[FIPS-186]等。

下面分别对所列的其中一些算法进行简要介绍。

## C.1　RIPEMD 算法

欧洲统一市场的集成宽带通信计划(IBC)于 1996 年提供商业应用。欧共体委员会为此有一个"欧洲先进通信技术研究发展计划"(Research and Development in Advanced Communication Technologies in Europe,RACE)。该计划中有一个子项目 RIPE(RACE Integrity Primitives Evaluation),其目标是采用现代密码学的方法提供一套用于保障信息系统中信息完整性的一些基本算法,以满足 IBC 的安全需求。

RIPEMD(RIPE Message Digest)是为 RIPE 项目研制的,它是 MD4 算法的一种变形,压缩函数共有 3 轮操作,产生 128 位的 Hash 值。它改变了循环移位和消息字的顺序,而且该算法的仅常数不同的两个实例是并行运行的,最后将两个实例的输出相加就得到了 RIPEMD 的 Hash 值。

1995 年,Hans Dobbertin 找到了两轮版本的 RIPEMD 的冲突,后来使用同样的技术,还找到了完整版本的 MD4 的冲突和 MD5 压缩函数的冲突。

随着对安全性要求的提高,出现了 RIPEMD-160 和 RIPEMD-128;因为在某些应用中需要更长的 Hash 值,又出现了 RIPEMD-256 和 RIPEMD-320。

RIPEMD-160 是 Hans Dobbertin、Antoon Bosselaers 和 Bart Preneel 于 1996 年提出的,性能与 SHA-1 相似,它没有申请专利,是开放的,但其应用没有 SHA-1 广,因而对它的分析也没有 SHA-1 多。它接收任意长输入,填充方法同 MD4,以 512-位分组进行处

理,对 5 个 32-位链接变量操作,最后连接产生 160 位的 Hash 值。压缩函数有 5 轮可并行的操作,每轮 16 步。它在选择参数上做得更好,而且两个运行实例中消息块的使用顺序完全不同、布尔函数的使用顺序也是相反的。对寄存器 A 的操作同 MD5,对寄存器 C 的环移操作增强了最高位的雪崩效应。在不同的轮中对消息字的使用并不仅仅是顺序不同,而是前面消息字之和。

与 RIPEMD-160 相比,RIPEMD-128 有 4 轮,操作中用到 4 个 32-位的链接变量,最后连接产生 128 位的 Hash 值,没有对寄存器 C 的环移操作。

RIPEMD-256 和 RIPEMD-320 分别是 RIPEMD-128 和 RIPEMD-160 的扩展版本。运行两个具有不同初始值的 RIPEMD-128 和 RIPEMD-160 实例,在每一轮执行后交换相应的一个链接变量,最后连接所有的链接变量就得到相应的双倍长度的 Hash 值。需要说明的是:它们只是减少了意外碰撞(accidental collision)的概率,其安全性并不比 RIPEMD-128 和 RIPEMD-160 强。

## C.2 HAVAL 算法

HAVAL 是 Yuliang Zheng、Josef Pieprzvk 和 Jennifer Seberry 于 1992 年提出的一个 MD5 的改进版本,它是一种 Hash 值长度可变的单向 Hash 算法。它以 1024 位为分组处理消息,有 8 个 32-位的链接变量,可以有 3、4、5 轮(每轮有 16 步操作),能产生长度为 128、160、192、224、256 位的 Hash 值。可变的轮数和可变的输出长度意味着该算法有 $3\times5=15$ 种不同的形式。

HAVAL 采用高非线性的 7-变量函数取代了 MD5 的简单非线性函数,且每一轮函数均能满足严格雪崩准则。每轮使用单个函数,但在每一步对输入进行不同的置换。该算法有一个新的消息次序,且每一步(第一轮除外)使用不同的加法常数。算法的核心是:

$$\begin{aligned}
TEMP &= (f(j,A,B,C,D,E,F,G) <<< 7) + (H <<< 11) + M[i][r(j)] + K(j);\\
& H = G;\ G = F;\ F = E;\ E = D;\\
& D = C;\ C = B;\ B = A;\\
& A = TEMP
\end{aligned}$$

在 2003 年亚洲密码学会议上,B. Van Rompay、A. Biryukov、B. Preneel 和 J. Vandewalle 以 $2^{29}$ 的复杂度找到了 3 轮 HAVAL 的一个冲突。2004 年亚洲密码学会议上冯登国等人又以 $2^{27}$ 的复杂度找到了 HAVAL-128 的冲突。

## C.3 SHA 算法

基于 MD4 和 MD5 的设计原则,1993 年美国标准技术研究院(NIST)和国家安全局(NSA)共同设计了 SHA-0(Secure Hash Algorithm)。1995 年,因为发现了一个未能公布的技术弱点,NSA 对 SHA 做了改进,这就是 SHA-1。随着安全性需求的增强,在此基础上又出现了 Hash 值更长的 SHA-224、SHA-256、SHA-384 和 SHA-512,它们被统称为 SHA-2。

## 1. SHA-0

(1) 算法描述

SHA-0(FIPS-180,Federal Information Processing Standard,联邦信息处理标准)接收任意长度小于 $2^{64}$ 位的输入,分组长度为 512 位,填充采用增强 MD 方法;每一分组划分为 16 个 32-位消息字,通过扩展算法变成供各步操作使用的 80 个消息字;算法用到 5 个 32-位链接变量,压缩处理后将它们级联形成一个 160-位的 Hash 值。

对消息分组的扩展操作如下:把消息分组从 16 个消息字(M0～M15)扩展到 80 个消息字(W0～W79):

$$Wt = Mt, 对\ t = 0\ 至\ 15$$

$$Wt = Mt-3 \oplus Mt-8 \oplus Mt-14 \oplus Mt-16, 对\ t = 16\ 至\ 79$$

压缩函数的处理分为 4 轮,每轮进行 20 步操作,各步的操作如下:

```
For t = 0 至 79
{
        TEMP = (a<<<5) + Φ(b,c,d) + e + Wt + Kt;
        e = d;
        d = c;
        c = b<<<30;
        b = a;
        a = TEMP
}
```

其中,函数 Φ 有 4 个,依次用在第 1～4 轮中:

```
F(X,Y,Z) = (X∧Y)∨((~X)∧Z)
G(X,Y,Z) = X⊕Y⊕Z
H(X,Y,Z) = (X∧Y)∨(X∧Z)∨(Y∧Z)
I(X,Y,Z) = X⊕Y⊕Z
```

每轮中都使用不同常数 $K_t = 2^{32} \times m^{\frac{1}{2}}/4$,其中,$m$ 依次取 2、3、5、10。

(2) 安全性

1995 年 NIST 和 NSA 就发现了 SHA-0 的一个安全性弱点,但是一直没有透露。

1998 年亚洲密码学会议上,Florent Chabaud 和 Antoine Joux 使用与分组密码的差分分析相关的方法,以 $2^{61}$ 的复杂度找到了 SHA-0 的冲突。

2004 年亚洲密码学会议上,Eli Biham 和 Rafi Chen 找到了 SHA-0 的一个近似冲突,其中两个消息的 160-位 Hash 值中,有 142 位是相同的,同时提出了通过近似冲突找到完全冲突的技术,以 $2^{29}$ 的复杂度找到了 65 轮的 SHA-0 的多个完全冲突;Joux、Carribault、Lemuet 和 Jalby 使用 Chabaud 和 Joux 的攻击技术,在 $2^{51}$ 的复杂度下找到了 SHA-0 的一个完全冲突;冯登国等人声称用他们的技术可以在 $2^{45}$ 的复杂度下找到 SHA-0 的冲突,但是没有提供实例。

2005 年 2 月 13 日,王小云等以 $2^{39}$ 的复杂度找到了一个 SHA-0 的冲突。

2．SHA-1

SHA-1(FIPS-180-1)作为 MD5 的继承算法，广泛应用于各种安全协议，例如 TLS、SSL、PGP、SSH、S/MIME 和 IPSec。它由 NSA 设计，是美国国家标准。

(1) 算法描述

SHA-1 与 SHA-0 的唯一区别在于：文件的预处理阶段有一个附加的操作，即消息分组的扩展中增加了移位操作，以消除前 20 轮的某些数学操作中存在的不可预知的安全隐患，具体如下：

Wt = Mt，对 t = 0 至 15

Wt = (Mt − 3 ⊕ Mt − 8 ⊕ Mt − 14 ⊕ Mt − 16) <<< 1，对 t = 16 至 49

SHA-1 的一次运算如图 C-1 所示。

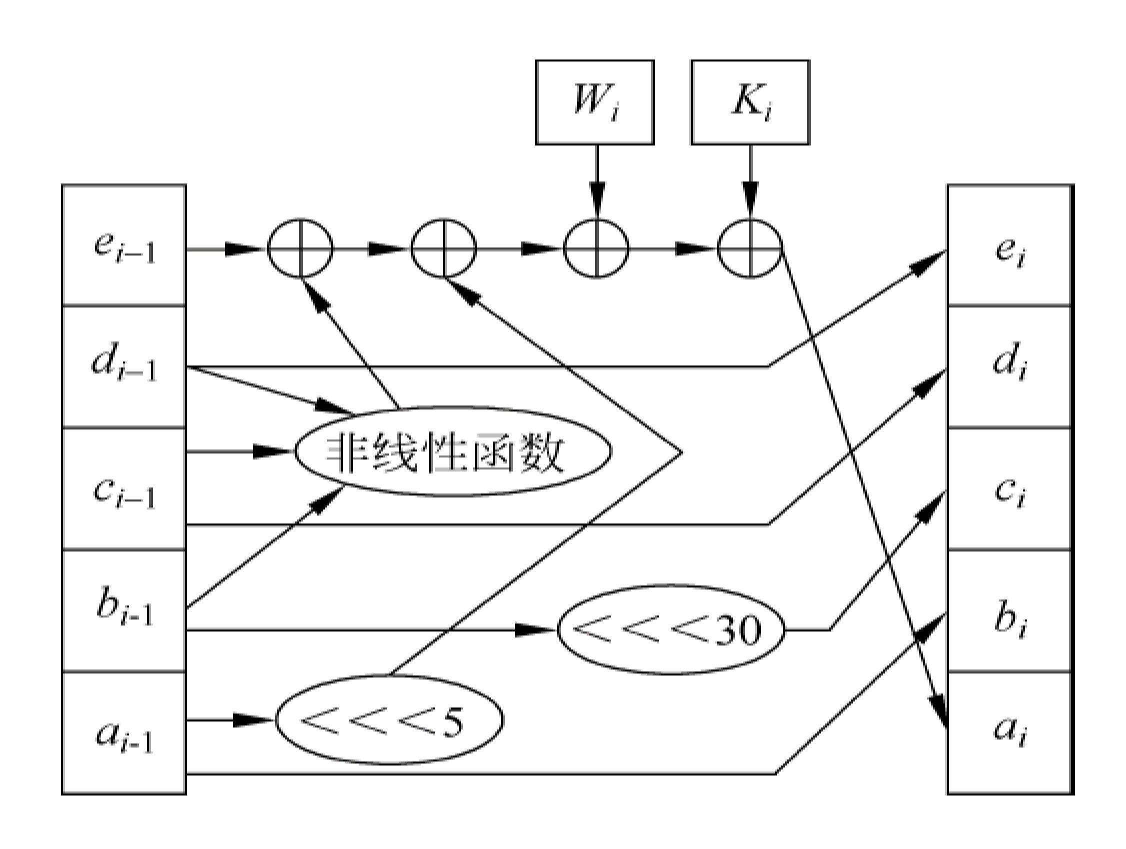



图 C-1 SHA-1 的一次运算

(2) 安全性

2005 年初，Rijmen 和 Oswald 以低于 $2^{80}$ 的复杂度找到了 53 轮 SHA-1 的一个冲突。2005 年 2 月，王小云与于红波、尹依群对 58 轮的 SHA-1，在 $2^{33}$ 的复杂度下找到实际的冲突；同时，首次提出能以低于 $2^{80}$ 的复杂度($2^{69}$)找到 SHA-1 的冲突。研究者声称，他们的分析基于攻击 SHA-0 使用的基本的差分分析技术、近似冲突技术，以及多块冲突技术、消息修改技术。主要利用了以下两个弱点：①文件的预处理不够复杂；②某些数学操作在前 20 轮中存在未预料到的安全问题。

2005 年 8 月的国际密码学会议上，公布了王小云、Andrew Yao 和 FranceYao 的最新成果，他们以 $2^{63}$ 的复杂度找到 SHA-1 的冲突。这样的复杂度已经可以通过大量分布的 Internet 搜索实现了，而且随着攻击技术的进一步发展，这个复杂度还可能降低，因此，SHA-1 也应该被淘汰了。

3．SHA-2

随着对 Hash 值长度要求的提高，2001 年又出现了 SHA-256、SHA-384 和 SHA-512，分别产生长度为 256、384 和 512 位的 Hash 值，它们被统称为 SHA-2。2002 年，它们和 SHA-1 一起成为官方标准 FIPS PUB 180-2。2004 年 2 月，SHA-224 又加入了这个

标准。

其中，SHA-384 和 SHA-512 的输入长度不超过 $2^{128}$ 位，分组长度为 1024 位；SHA-224 和 SHA-256 的输入长度不超过 264 位，分组长度为 512 位。

SHA-256 与 SHA-1 在结构上很相似，但是它用到 8 个 32-位的链接变量，共有 64 步操作，扩展算法和压缩函数都与 SHA-1 不同；SHA-224 基于 SHA-256，有不同的初始值，结果取前 224 位。

SHA-512 有与 SHA-256 相似的扩展算法和压缩算法，用到 8 个 64-位的链接变量，分组长度为 1024 位，共 80 步操作；SHA-384 基于 SHA-512，但是有不同的初始值，而且结果取前 384 位。

2003 年，Gilbert 和 Handschuh 研究了它们的安全性，没有发现任何弱点。2005 年以来，对它们的分析工作已经展开，但到目前为止，仍没有找到对这些 Hash 算法的有效攻击，它们靠着足够长的 Hash 值，将在今后几年中作为 SHA-1 的替代方案，直到新的标准出现。

## C.4　Whirlpool 算法

2000 年，Vincent Rijmen 和 Paulo S. L. M. Barreto 设计了 Whirlpool（M51）算法，它是目前 NESSIE（New European Schemes for Signature Integrity and Encryption）唯一推荐使用的 Hash 算法，同时它也被国际标准组织 ISO（International Organization for Standardization）和国际电子技术协会 IEC（International Electrotechnical Commission）采用作为 ISO/IEC 10118-3 国际标准。

Whirlpool 是在分组密码 Square 的基础上设计的，算法的输入长度不超过 $2^{256}$ 位，产生 512 位的 Hash 值。

最初的版本中，S-盒是随机生产的，具有良好的密码学特性。2001 年的版本中，对它进行了改进，使其密码学特性更好，而且更方便硬件实现；2003 年的版本中，进一步修改了扩散阵列（diffusion matrix）。

消息分组的长度为 512 位，填充同 MD5，其中，长度消息占最后 256 位，链接变量的初始值 h0 置为全 0，用到专门的压缩函数 W，使用 Miyaguchi-Preneel 方法构建压缩函数：hi＝Whi-1mi⊕mi⊕hi-1（1＜i＜t），hi 就是压缩函数的输出。W 类似于 Rijndael，与之相比有以下不同之处：分组长度固定、轮数固定、密钥扩展用的是轮函数、用到的归约多项式稍有不同；S-盒不同；轮常量不同；扩散层不同。

设计者声称此 Hash 算法能抗线性分析和差分分析，而且取 512 位 Hash 值中的任意位数都能保证达到相应的单向性和抗冲突性。

## C.5　Tiger 算法

1996 年，Ross Anderson 和 Eli Biham 提出了一个能在 64-位平台上有效运行的 Tiger 算法，它产生 192 位的 Hash 值。分组长度为 512 位，填充同 MD5，其中长度消息占最后的 64 位，有 4 个 64-位链接变量。算法的核心是三轮操作和各轮之间的密钥表，每轮有 8 步，涉及的操作除了 64-位的加法、减法和逻辑运算外，还有 64-位乘法运算和查 S-

盒的操作(将 8 位的输入映射为 64 位的输出),与其他算法相比,产生了更好的雪崩效应,而且消息中一个位的改变会影响很多位,这样能抵抗所有的差分分析(差分分析基于如下事实:改变消息字中一些特定位在许多轮中最多只影响链接变量的几个位,而且这些小的差分能在随后的轮操作中得到纠正)。

为了能与其他常用 Hash 算法兼容,也有 Tiger-160 和 Tiger-128,它们在 Tiger-192 算法的基础上,分别取前 160 位和 128 位作为最后的 Hash 值。

因为现有攻击都用到差分分析,所以,Tiger 目前还是安全的。

## C.6 MDC-2 和 MDC-4 算法

MDC-2(modification detection code 2)和 MDC-4(modification detection code 4)算法首先由 IBM 开发,它们都是基于分组密码的双倍长度 Hash 算法。MDC-2,有时称为 Meyer-Schilling,已考虑作为 ANSI 和 ISO 标准。MDC-4 是为 RIPE 项目研制的,尽管理论上可以使用任何加密算法,但 MDC-4 还是用 DES 作为分组函数。

MDC-2 的散列率为 1/2,产生的 Hash 值的长度为分组长度的两倍,如图 C-2 所示。

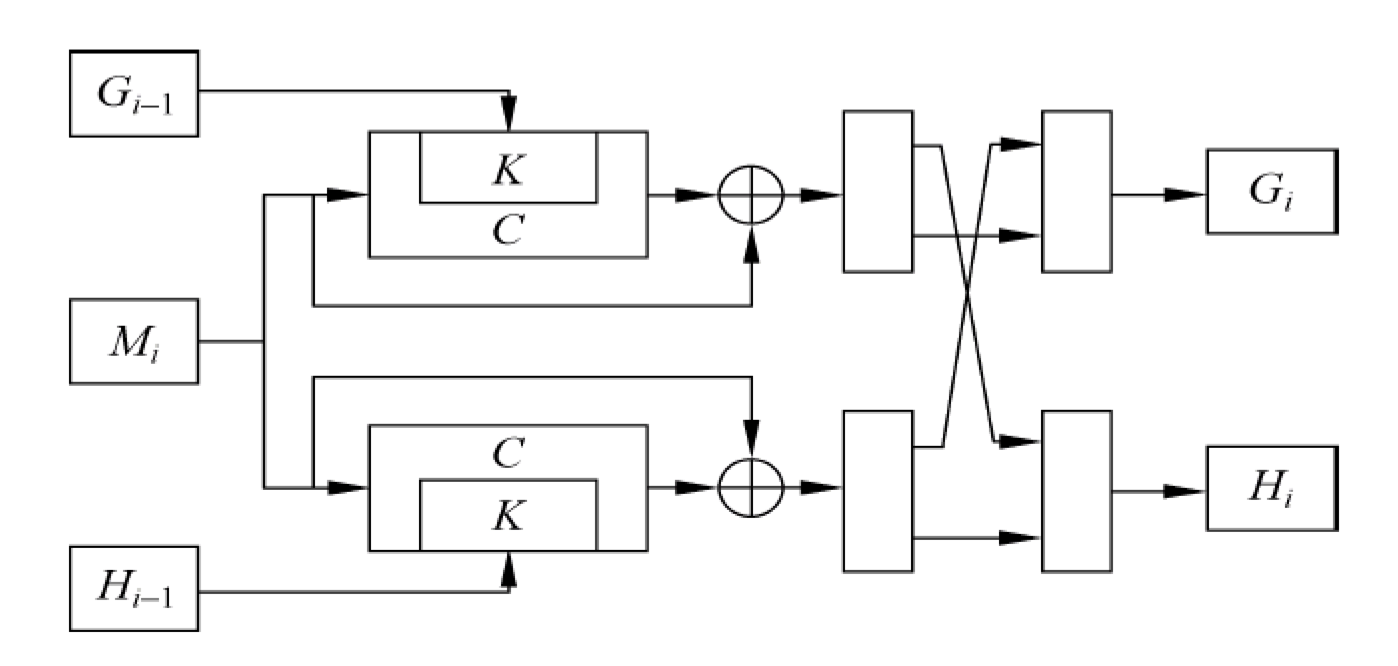



图 C-2 MDC-2

MDC-4 产生的 Hash 值的长度也是分组长度的两倍,其散列率为 1/4,如图 C-3 所示。

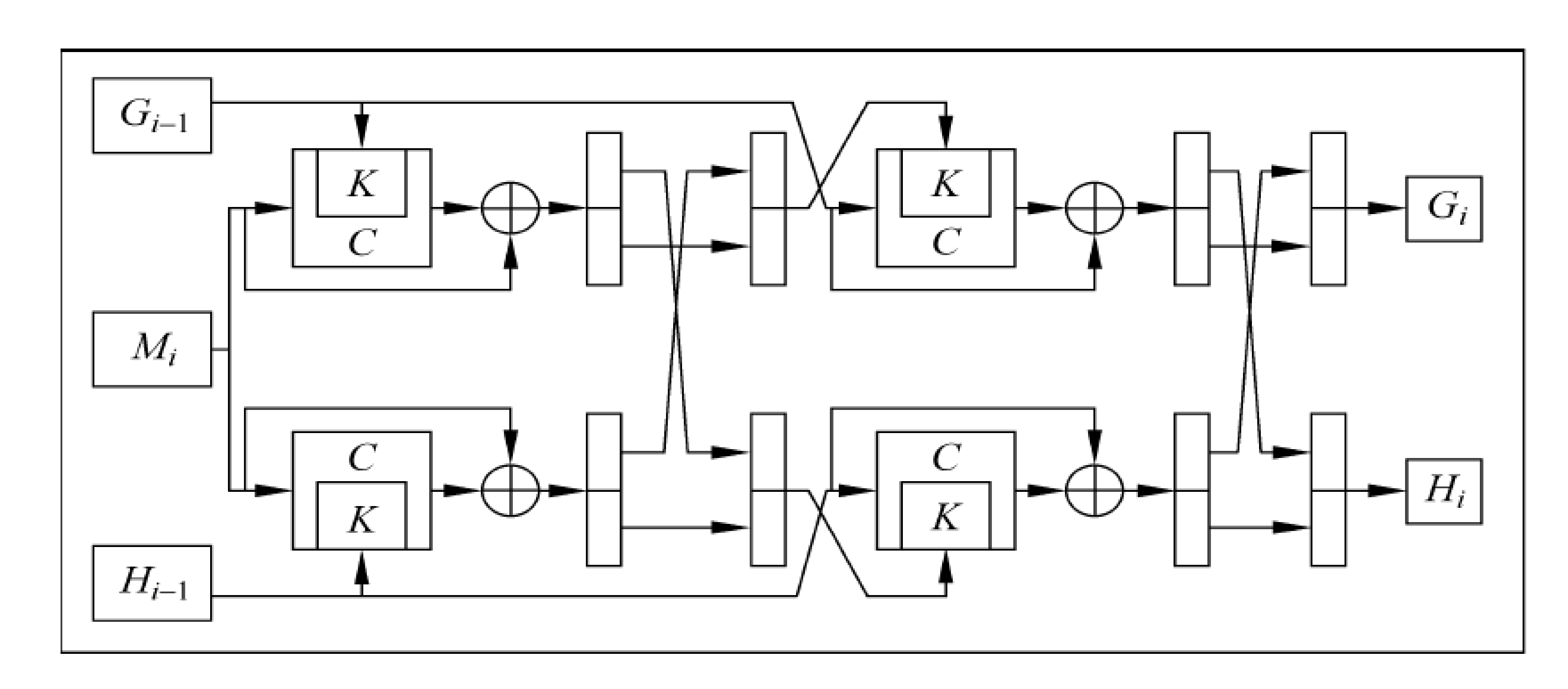



图 C-3 MDC-4

Knudsen 和 Preneel 攻破了 MDC-4,MDC-2 虽然还是没有被攻破,但执行效率很低。

# D　X.509 简介

X.509 是一种行业标准。目前 X.509 有不同版本，X.509v2 和 x.509v3 是比较新的版本，都是在原有版本（X.509v1）的基础上进行功能的扩充，其中每一版本都包含下列信息：

（1）版本号：用来区分 X.509 的不同版本号。

（2）序列号：由 CA 给予每一个证书分配唯一的数字型编号。当证书取消时，实际上是将此证书的序列号放入由 CA 签发的 CRL 中，这也是序列号唯一的原因。

（3）签名算法标识符：用来指定用 CA 签发证书时所使用的签名算法。算法标识符用来指定 CA 签发证书时所使用的公开密钥算法和 Hash 算法，需向国际指明标准组织（如 ISO）注册。

（4）认证机构：即发出该证书的机构唯一的 CA 的 x.500 名字。

（5）有效期限：证书有效的时间包括两个日期：证书开始生效期和证书失效的日期和时间。在所指定的这两个时间之间有效。

（6）主题信息：证书持有人的姓名、服务处所等信息。

（7）认证机构的数字签名：以确保这个证书在发放之后没有被篡改过。

（8）公钥信息：包括被证明有效的公钥值和加上使用这个公钥的方法名称。

## D.1　X.509 证书结构简介及实例

### 1. ASN1 简介

ASN1（abstract syntax notation one），抽象语法符号 1 是用来描述数据结构及其编码的规则集，它广泛用于 RFC（request for comments）文本中数据结构的描述。X.509 证书结构（RFC 2459）就是采用 ASN1 描述和编码的。因此首先对 ASN1 的编码规则做一简单介绍。

在 ASN1 编码中，数据类型包括简单类型和结构类型。简单类型是不能再分解类型，如整型（integer）、比特串（bit string）、字节串（octet string）、对象标示符（object identifier）、日期型（UTC time）等。结构类型是由简单类型和结构类型组合而成的，如顺序类型（sequence，sequenceof）、选择类型（choice）、集合类型（set）等。顺序类型的值由按给定顺序成员数据值组成；选择类型的值由多个成员数据类型中的某一个值构成；集合数据类型由成员数据类型的一个或多个值构成。每一种类型都有一个整数标记（TAG）来标识该类型。

不论是简单类型还是结构类型的值其编码都是由 4 部分构成：类型标识字段，用来标识该值的类型；长度字段，用来标识该值所占的字节数；值字段；结束标示字段。

在类型标识字段中，使用类型的 TAG 来标识该类型，在证书中使用一个字节来表示。bit8-bit7 用来标示 TAG 类型，bit6 标示是否为结构类型（1 位结构类型），bit5-bit1 是类型的 TAG 值。如 Sequence 类型，其 TAG 类型位 Universal（00），属于结构类型（1）；

TAG 值为 16(10000),所以其类型标示字段值为(00110000),即为 0x30。

长度字段有两种编码格式。若长度值小于等于 127,则用一个字节表示,bit8=0,bit7-bit1 存放长度值;若长度值大于 127,则用多个字节表示,第一个字节存放长度字段所占的字节数(<=127),并且 bit8=1,其余字节存放长度值。如果长度值不定,用一个字节(0x80)表示。

值字段,存放数据值,具体编码随值的数据类型不同而不同。

结束标示字段,两个字节(0x0000),只有在长度值为不定时才会出现。

### 2. X.509 证书结构

```
Certificate::= SEQUENCE{
tbsCertificateTBSCertificate,——证书主体
signatureAlgorithmAlgorithmIdentifier,——证书签名算法标识
signatureValueBITSTRING——证书签名算法值
}

TBSCertificate::= SEQUENCE{
version[0]EXPLICITVersionDEFAULTv1,——证书版本号
serialNumberCertificateSerialNumber,——证书序列号,对同一 CA 所颁发的证书,序列号唯一标
识证书
signatureAlgorithmIdentifier,——证书签名算法标识
issuerName,——证书发行者名称
validityValidity,——证书有效期
subjectName,——证书主体名称
subjectPublicKeyInfoSubjectPublicKeyInfo,——证书公钥
issuerUniqueID[1]IMPLICITUniqueIdentifierOPTIONAL,——证书发行者 ID(可选),只在证书版本
2、3 中才有
subjectUniqueID[2]IMPLICITUniqueIdentifierOPTIONAL,——证书主体 ID(可选),只在证书版本
2、3 中才有
extensions[3]EXPLICITExtensionsOPTIONAL——证书扩展段(可选),只在证书版本 2、3 中才有
}

Version::= INTEGER{v1(0),v2(1),v3(2)}
CertificateSerialNumber::= INTEGER

Validity::= SEQUENCE{
notBeforeTime,——证书有效期起始时间
notAfterTime}——证书有效期终止时间

Time::= CHOICE{
utcTimeUTCTime,
generalTimeGeneralizedTime}

UniqueIdentifier::= BITSTRING

SubjectPublicKeyInfo::= SEQUENCE{
algorithmAlgorithmIdentifier,——公钥算法
```

```
subjectPublicKeyBITSTRING}——公钥值

Extensions::= SEQUENCESIZE(1..MAX)OFExtension

Extension::= SEQUENCE{
extnIDOBJECTIDENTIFIER,
criticalBOOLEANDEFAULTFALSE,
extnValueOCTETSTRING}
```

### 3. 证书实例

本部分包含一个699字节的证书实例。证书版本号为3。该证书包含以下内容：

(1) 证书序列号是17(0x11)。

(2) 证书使用DSA和SHA-1哈希算法签名。

(3) 证书发行者的名字是OU=nist;O=gov;C=US。

(4) 证书主体的名字是OU=nist;O=gov;C=US。

(5) 证书的有效期从1997-6-30到1997-12-31。

(6) 证书包含一个1024bit DSA公钥及其参数。

(7) 证书包含一个主体键标识扩展项。

(8) 证书是一个CA证书(通过一个基本扩展项标识)。

## D.2　X.509的扩展(v3)

X.509标准第三版在v2的基础上进行了扩展，v3引进一种机制。这种机制允许通过标准化和类的方式将证书进行扩展包括额外的信息，从而适应下面的一些要求：

(1) 一个证书主体可以有多个证书。

(2) 证书主体可以被多个组织或社团的其他用户识别。

(3) 可按特定的应用名(不是X.500名)识别用户，如将公钥同e-mail地址联系起来。

(4) 在不同证书政策和实用下会发放不同的证书，这就要求公钥用户要信赖证书。

证书并不限于这些标准扩展，任何人都可以向适当的权利机构注册一种扩展。将来会有更多的适于应用的扩展列入标准扩展集中。值得注意的是：这种扩展机制应该是完全可以继承的。

每一种扩展包括3个域：类型、可否默认、扩展值。

(1) 类型字段定义了扩展值字段中的数据类型。这个类型可以是简单的字符串，数值，日期，图片或一个复杂的数据类型。为便于交互，所有的数据类型都应该在国际知名组织进行注册。

(2) 可否默认字段是一比特标识位。当一扩展标识为不可默认时，说明相应的扩展值非常重要，应用程序不能忽略这个信息。如果使用一特殊证书的应用程序不能处理该字段的内容，就应该拒绝此证书。

(3) 扩展值字段包含这个扩展实际的数据。

公开密钥证书的标准扩展可以分为以下几组：

（1）密钥和政策信息：包括机构密钥识别符，主体密钥识别符，密钥用途（如数字签字，不可否认性、密钥加密、数据加密、密钥协商、证书签字、CRL 签字等），密钥使用期限等。

（2）主体和发证人属性：包括主体代用名、发证者代用名、主体检索属性等。

（3）证书通路约束：包括基本约束，指明是否可以做证书机构。

## D.3 CRL 和 CRL 扩展简介

CRL（certificate revocation list）—— 证书废弃列表，它是盖了时间印章又经过 CA 签名，自由发布长期有效并能识别出被撤销证书的清单。下面对 CRL 的介绍都来源于文档 RFC 2459，在此只列出部分内容，具体可参考该文档所述。

CRLs 可以用在覆盖宽广范围的可由双方共同操作目标，甚至用在更宽广范围的应用和环境中的操作上，并能保证需求。在该简介中为适应范围宽广可由双方共同操作的应用建立一条共用基线。为基线定义一套能在每一个 CRL 中被预期的信息。同时，也定义这些属性（共用代表经常用到的属性）在 CRL 以内的通用位置。

根据附加或者特殊目的需求，环境可以建立在该简介基础上或者可以取代它。如果其他撤销或者提供证书状况算法，遵照 CAs，不需要发行 CRLs。发行 CRLs 的 CAs 必须发行第 2 版 CRLs，CAs 必须在 nextUpdate 字段中包含日期（下一个 CRL 发出的日期），CRL 号扩展和权威密钥标识扩展。按照应用的需要，处理第 1 版和第 2 版 CRLs。

### 1. CRL 字段

X.509v2 CRL 句法如下。为签名计算，要签名的数据是 ASN.1 DER 编码。ASN.1 DER 编码是标签、长度、对应每一元素值的编码系统。

```
CertificateList ::= SEQUENCE {
      tbsCertList            TBSCertList,
      signatureAlgorithm     AlgorithmIdentifier,
      signatureValue         BIT STRING  }

TBSCertList  ::=  SEQUENCE  {
      version                    Version OPTIONAL,
                                      -- if present, shall be v2
      signature                  AlgorithmIdentifier,
      issuer                     Name,
      thisUpdate                 Time,
      nextUpdate                 Time OPTIONAL,
      revokedCertificates        SEQUENCE OF SEQUENCE  {
           userCertificate            CertificateSerialNumber,
           revocationDate             Time,
           crlEntryExtensions         Extensions OPTIONAL
                                          -- if present, shall be v2
                                      }  OPTIONAL,
      crlExtensions              [0] EXPLICIT Extensions OPTIONAL
                                          -- if present, shall be v2
                                      }
```

(1) CertificateList 字段

CertificateList 是一连串(sequence)的 3 个需求字段(three required fields)。字段在下边详细描述。

① tbsCertList。在序列中第一字段是 tbsCertList。这个字段表示序列自身,含有发行者的名字、发行日期,下一个清单(CRL)的发行日期,撤销证书列表和可选 CRL 扩展。进一步,在撤销证书清单上每个入口是由一串用户证书确定序号,撤销日期和可选 CRL 入口扩展。

② signature algorithm 字段。signature algorithm 字段为含有算法标识符,经过 CA 在 CertificateList 上签名使用的算法。

③ signature value 字段。signature value 字段在 ASN.1 DER 编码 tbsCertList 上含有一个数字签名。使用 ASN.1 DER 编码 tbsCertList 作为签名函数的输入。

(2) 证书列表"To Be Signed"

签名的证书清单或者 TBSCertList 是一串必须和可选的字段。必须字段标识 CRL 发行者、用在 CRL 上签名的算法、CRL 发出的日期和时间以及在 CA 将发行下一个 CRL 的日期和时间。

可选字段包含撤销证书和 CRL 扩展的清单。

① 版本。这个可选字段描述编码 CRL 的版本。

② 签名。这个字段含有用在 CRL 上签名算法的算法标识符。

③ 发行者名字。发行者名字标识已经在 CRL 上签名和发行的实体。在发行者名字字段中带有发行者实体。可替换名字格式也可以出现在 Issuer Alt Name 扩展中。

④ 更新。这个字段指示 CRL 的发布日期。ThisUpdate 可以作为 UTCTime 或者 GeneralizedTime 被编码。

⑤ 下次更新。该字段指示下一个 CRL 发出的日期。

⑥ 撤销证书。列举撤销证书。通过他们的序号给撤销证书命名。通过证书序列号唯一标识被撤销的证书。指定撤销发生的日期。Revocation date 时间必须是如同在该小节的部分④中描述的那样表示。

⑦ 扩展。这个字段仅出现在版本 2(或者更高版本中)。如果存在,这个字段是一串(Sequence)的一或更多 CRL 扩展。CRL 扩展在下面讨论。

## 2. CRL 扩展

由 ANSI X9 和 ISO/IEC/ITU 为 X.509v2 CRLs X.509 定义扩展。X9.55 提供把附加属性和 CRLs 联系起来的方法。X.509v2 CRL 格式也允许社区定义私有范围携带那些社区所特有信息的扩展。在 CRL 中每一个扩展可以是关键或者非关键的。如果它遇到一种它不知道如何处理的关键扩展,CRL 肯定(MUST)失败。但是,未认出的非关键扩展可以忽视。以下部分提出在 InternetCRLs 以内使用的扩展。社区可以选择在本文说明中包含在 CRLs 中不定义的扩展。但是,可以在一般背景方面运用的 CRLs 中应该谨慎采用任何关键扩展。

发行 CRLs 的 CAS 确认包含权威密钥标识符和在所有发行的 CRLs 中的 CRL 数字

扩展。

(1) 权威密钥标识符

权威密钥标识符扩展提供一种识别出对使用私有密钥在CRL上签名相对应的公开密钥的手段。确认能建立在任一个基础上密钥标识符(在CRL签名者的证书中的主题密钥标识符)或者有关发行者名字和序号。一个发行者有超过一个由于多重同时发生的密钥对或者由于密钥对更换的地方,这扩展特别有用。

CAS必须使用密钥标识符方法,必须在所有发行的CRLs中包含这个扩展。

(2) 发行者可替换名字

发行者可替换名字扩展允许额外标识和CRL的发行者联系起来。定义可选项包含RFC 822名字(电子邮件地址),DNS名字,IP地址和URI。可以包含名字和多重名字格式的多重事例。每当使用这样的标识符时,必须使用发行者可替换名字扩展。Issuer Alt Name扩展不应该标记为关键。

(3) CRL Number

CRL Number是经过CA发出的非关键CRL扩展,其为每一个CRL表达一种单调增加的序列号。这个扩展允许用户容易决定什么时候一个特定CRL取代另一个CRL。CAS必须在所有CRLs中包含该扩展。

(4) Delta CRL标识符

Delta CRL标识符是识别出一个delta-CRL的CRL关键扩展。使用delta-CRL能在相当大程度上为应用(程序)提高在CRL结构之外储藏撤销信息格式的处理时间。这就允许把改变加入本地数据库,同时忽视未改变的信息(其已经在本地数据库中)。

当发出一个delta-CRL的时候,CAS必须同时发行一个完整的CRL。BaseCRLNumber的值标识的CRL号是在产生delta-CRL时的起始点。

(5) 发行发布点

发行发布点是一个关键CRL扩展,为一个特定CRL识别出CRL发布点,并且它指示CRL是否仅为末端实体证书、仅为CA证书或者一套限制(limitied)理由代码覆盖撤销。虽然扩展是关键,并不保证工具是支持这个扩展所需要的。

### 3. CRL入口扩展

已经由ANSI X9和ISO/IEC/ITU为X.509v2CRLs定义的CRL入口扩展提供为附加属性和CRL入口[X.509][X9.55]联系起来的方法。X.509v2CRL格式也允许社区定义私有CRL入口扩展携带那些社区所特有信息。每一个在CRL入口中的扩展可以是关键或者非关键的。如果遇到一个它不知道如何处理的关键CRL入口扩展,CRL批准肯定失败。但是,一个未认出的非关键CRL入口扩展可以忽视。以下部分提出在Internet CRL入口和标准信息的位置以内使用推荐的扩展。社区可以选择使用附加CRL入口扩展,但是,在采用任何可以用在一般背景中CRL的入口中的关键扩展应该谨慎。

在该简介中使用的所有CRL入口扩展都是非关键的。支持这些扩展,因为对CAs和应用保证是可选的。但是,每当这些信息是可用的,发行CRLs的CAs应该(Should)

包含理由代码和无效日期。

(1) 理由代码

Reasoncode 是一个非关键 CRL 入口扩展,其识别证书撤销的理由。CAs 强烈鼓励在 CRL 入口中包含有意义的理由代码。

(2) 保持指示代码

保持指示代码是一个非关键 CRL 入口扩展,其提供一个登记指示标识符,其指示行动要在遇到一张证书之后,并放在着力点(hold)上。

(3) 无效日期

无效日期是一个非关键 CRL 入口扩展,其定义日期可知道或者怀疑其私有密钥或者证书被泄露,除此之外是无效的。

(4) 证书发行者

该 CRL 入口扩展识别出与一个间接 CRL 入口联系起来的证书发行者(间接 CRL 是指:如果含有 Indirect CRL,被指示者嵌入它的发行发布点扩展)。如果这个扩展在一间接 CRL 中第一入口点上不存在,证书发行者默认 CRL 发行者;在一间接 CRL 中的随后入口点上,如果该扩展不存在,对入口的证书发行者和那个前面入口一样。

# 参 考 文 献


[1] William Stallings. Cryptography and Network Security Principles and Practice. Prentice-Hall Inc. , 1999

[2] Karanjit S, Chirs H. Internet Firewall and Network Security. New Riders Publishing, 1990

[3] Steven Brown. Implementing Virtual Private Networks. New York: McGraw-Hill, 1999

[4] Rebecca Gurley Bace. Intrusion Detection. U. S. A. : Macmillan Technical Publishing, 1999

[5] Martin Roesch. Snort lightweight intrusion detection for networks. In: The Proceedings of the 13th Large Installation System Administration Conference, Seattle, Washington, USA, 1999

[6] Web ST and Network Security. DASCOM(Holdings)Ltd, 1997

[7] Marc Farley, Tom Stearns and Jeffrey Hsu. LAN Times Guide to Security and Data Integrity. Copyright 1996 by McGraw-Hill, Inc

[8] Ford. Computer Communications Security. PTR Prentice Hall, Englewood Cliffs, New Jersey, 1994

[9] Ribenboim P. The New Book of Prime Number Records. New York: Springer-Verlag, 1996

[10] Adams C. Simple and Effective Key Scheduling for Symmetric Ciphers. Proceedings, Workshop in Selected Areas of Cryptography, SAC'94, 1994

[11] Barker W. Introduction to the Analysis of the Data Encryption Standard (DES). Laguna Hills, CA: Aegean Park Press, 1991

[12] Bellovin S and Merritt M. Limitations of the Kerberos Authentication System. Computer Communications Review, October 1990

[13] Bellovin S and Cheswick W. Network Firewalls. IEEE Communications Magazine, September 1994

[14] Bellare M, Canetti R and Krawczyk H. Keying Hash Functions for Message Authentication. Proceedings, CRYPTO'96, August 1996, New York: Springer-Verlag. An expanded version is available at http://www.cse.ucsd.edu/user/mihir

[15] Berson T. Differential Cryptanalysis Mod $2^{32}$ with Applications to MD5. Proceedings, EUROCRYPT'92, May 1992, New York: Springer-Verlag

[16] Beth T, Frisch M and Simmons G eds. Public-Key Cryptography: State of the Art and Future Directions. New York: Springer-Verlag, 1991

[17] Biham E and Shamir A. Differential Cryptanalysis of the Data Encryption Standard. New York: Springer-Verlag, 1993

[18] Biham E and Shamir A. Power Analysis of the Key Scheduling of the AES Candidates. Proceedings, Second AES Candidate Conference, 24 October 2000. http://csrc.nist.gov/encryption/aes/round1/conf2/aes2conf.htm

[19] Boer B and Bosselaers A. Collisions for the Compression Function of MD5. Proceedings, EUROCRYPT'93, 1993, New York: Springer-Verlag

[20] Chapman D and Zwicky E. Building Internet Firewalls. Sebastopol, CA: O'Reilly, 1995

[21] Cheng P, et al. A Security Archtecture for the Internet Protocol. IBM Systems Journal, Number 1, 1998

[22] Cheswick W and Bellovin S. Firewalls and Internet Security: Repelling the Wily Hacker. Reading, MA: Addison-Wesley, 2000

[23] Daemen J and Rijmen V. AES Proposal: Rijndael, Version 2. Submission to NIST, March 1999. http://csrc.nist.gov/encryption/aes
[24] Davies D and Price W. Security for Computer Networks. New York: Wiley, 1989
[25] Denning D. Timestamps in Key Distribution Protocols. Communications of the ACM, August 1981
[26] Denning D. Cryptograhy and Data Security. Reading, MA: Addison Wesley, 1982
[27] Diffie W and Hellman M. New Directions in Cryptography. Proceedings of the AFIPS National Computer Conference, June 1976
[28] Diffie W and Hellman M. Multiuser Cryptographic Techniques. IEEE Transactions on Information Theory, November 1976
[29] Diffie W and Hellman M. Privacy and Authentication: An Introduction to Cryptography. Proceedings of the IEEE, March 1979
[30] Drew G. Using SET for Secure Electronic Commerce. Upper Saddle River, NJ: Prentice Hall, 1999
[31] Doraswamy N and Harkins D, IPSec. Upper Saddle River, NJ: Prentice Hall, 1999
[32] ElGamal T. A Public-Key Cryptosystem and a Sinature Scheme Based on Discrete Logarithms. IEEE Transactions on Information Theory, July 1985
[33] Enger N and Howerton P. Computer Security. New York: Amacom, 1980
[34] Feistel H, Notz W and Smith J. Some Cryptographic Techniques For Machine-To-Machine Data Communications. Proceedings of the IEEE, November 1975
[35] Ford W. Advances in Public-Key Certificate Standards. ACM SIGSAC Review, July 1995
[36] Frankel S. Demystifying the IPSec Puzzle. Boston: Artech House, 2001
[37] Fumy S and Landrock P. Principles of Key Management. IEEE Journal on Selected Areas in Communications, June 1993
[38] Gardner M. Codes, Ciphers and Secret Writing. New York: Dover, 1972
[39] Garfinkel S and Spafford G, Web Security & Commerce. Cambridge, MA: O'Reilly and Associates, 1997
[40] Gollmann D. Computer Security. New York: Wiley, 1999
[41] Gong L. A Security Risk of Depending on Synchronized Clocks. Operating Systems Review, January 1992
[42] Abadi M, Tuttle M R. A Semantics For A Logic Of Authentication. In: Proceedings of the 10th ACM Symposium on Principles of Distributed Computing. ACM Press, 1991, 201～216
[43] Asokan N, Shoup V, Waidner M. Asynchronous protocols for optimistic fair exchange, In: Proceedings of the 1998 IEEE Symposium on Security and Privacy. Los Alamitos: IEEE Computer Society Press, 1998, 86～89
[44] Bellare M, et al. iKP-A Family of Secure Electronic Payment Protocols. IBM Research Division, Zurich Research Lab, July 1995
[45] Burrows M, Abadi M, Needham R. Rejoinder to Nessett. Operating Systems Review, 1990, 24(2): 39～40
[46] Burrows M, Abadi M, Needham R. A logic of authentication. Research report 39, digital systems research center. February 1989. In: Proceedings of the Royal Society of London A, 1989, 426: 233～271
[47] Clark J, Jacob J. A survey of authentication protocol literature: Version 1.0. Available at www-users.cs.york.ac.uk/～jac/under the link\Secure Protocols Review, 1997

[48] Coffey T,Saidha P. Non-repudiation with mandatory proof of receipt. Computer Communication Review. 1996,26(1): 6～17
[49] Deng R H,Gong L,Lazar A,Wang W. Practical protocols for certified electronic mail. Journal of Network and Systems Management,1996,(3): 279～297
[50] Diffie W, Hellman M E. New Directions In Cryptography. IEEE Transactions on Information Theory,1976,IT-22(6): 644～654
[51] Dolev D, Yao A. On the Security of Public Key Protocols. IEEE Transactions on Information Theory,1983,29(2): 198～208
[52] Frier A, Karlton P, Kocher P. The SSL 3. 0 Protocol. Netscape Communication Corp. Nov 18,1996
[53] Gong L, Needham R, Yahalom R. Reasoning about belief in cryptographic protocols. In: Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy. Los Alamitos: IEEE Computer Society Press,1990,234～248
[54] Gong L. Cryptographic Protocols for Distributed Systems. Ph. D. thesis. University of Cambridge. April 1990
[55] Guttman J D,Thayer F J. Authentication tests. In: Proceedings of the 2000 IEEE Symposium on Security and Privacy,Los Alamitos: IEEE Computer Society Press,2000,150～164
[56] Horng G,Hsu C K. Weakness in the Helsinki protocol. Electronic Letters,1998,34: 354～355
[57] Kailar R. Reasoning about accountability in protocols for electronic commerce. In: Proceedings of the IEEE Symposium on Security and Privacy, Los Alamitos: IEEE Computer Society Press, 1995,236～250
[58] Lowe G. Breaking and fixing the Needham-Schroeder public～key protocol using FDR. Software-Concepts and Tools,1996,17: 93～102
[59] Meadows C. The NRL Protocol Analyzer: An overview. Journal of Logic Programming,1996, 26(2): 113～131
[60] Otway D,Ress O. Efficient and Timely Mutual Authentication. Operating Systems Review. 1987, 21(1): 8～10
[61] Rubin A D. Nonmonotonic Cryptographic Protocols. PhD thesis, University of Michigan, Ann Arbor,1994
[62] Schneider S A. Using CSP for protocol analysis: the Needham-Schroeder Public-Key Protocol, Technical Report CSD-TR-96-14,Royal Holloway,University of London,1996
[63] Song D. Athena. A New Efficient Automatic Checker For Security Protocol Analysis. In: Proceedings of the 1999 IEEE Computer Security Foundations Workshop, Los Alamitos: IEEE Computer Society Press,1999,192～202
[64] Syverson P F, Van Oorschot P C. On Unifying Some Cryptographic Protocol Logics. In: Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy. Los Alamitos: IEEE Computer Society Press,1994,14～28
[65] Thayer F J,Herzog J C,Guttman J D. Strand spaces: Proving security protocols correct. Journal of Computer Security,1999,7(2,3): 191～230
[66] Van Oorschot P C. Extending cryptographic logics of belief to key agreement protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security. ACM Press, 1993,233～243
[67] Woo T,Lam S. A Lesson on Authentication Protocol Design. Operating Systems Review,1994, 24～37

[68] Zhou J, Gollmann D. Towards verification of non-repudiation protocols. In: International Refinement Workshop and Formal Methods Pacific 1998, Berlin: Springer-Verlag, 1998, 370~380
[69] Gong L. Variations on the Themes of Message Freshness and Replay. Proceedings, IEEE Computer Security Foundations Workshop, June 1993
[70] Hevia A and Kiwi M. Strength of Two Data Encryption Standard Implementations Under Timing Attacks. ACM Transactions on Information and System Security, November 1999
[71] Heys H and Tavares S. Avalanche Characteristics of Substitution Permutation Encryption Networks. IEEE Transactions on Computers, September 1995
[72] Huitema C. IPv6: The New Internet Protocol. Upper Saddle River, NJ: Prentice Hall, 1998
[73] I'Anson C. and Mitchell C. Security Defects in CCITT Recommendation X. 509——The Directory Authentication Framework. Computer Communications Review, April 1990
[74] Javitz H and Valdes A. The SRI IDES Statistical Anomaly Detector. Proceedings, 1991 IEEE Computer Society Symposium on Research in Security and Privacy, May 1991
[75] Jones R. Some Techniques for Handling Encipherment Keys. ICL Technical Journal, November 1982
[76] Jueneman R, Matyas S and Meyer C. Message Authentication. IEEE Communications Magazine, September 1998
[77] Jueneman R. Electronic Document Authentication. IEEE Network Magazine, April 1987
[78] Kaliski B and Robshaw M. The Secure Use of RSA. CryptoBytes, Autumn 1995
[79] Kaliski B and Robshaw M. Multiple Encryption: Weighing Security and Performance. Dr. Dobb's Journal, January 1996
[80] Kehne A, Schonwalder J and Langendorfer H. A Nonce-Based Protocol for Multiple Authentications. Operating Systems Review, October 1992
[81] Koblas D and Koblas M. SOCKS. Proceedings, UNIX Security Symposium III, September 1992
[82] Kocher P. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. Proceedings, Crypto'96, August 1996
[83] Kohnfelder L. Towards a Practical Public-Key Cryptosystem. Bachelor's Thesis, M. I. T., May 1978
[84] Kohl J. The Use of Encryption in Kerberos for Network Authentication. Proceedings, Crypto '89, 1989, New York: Springer-Verlag
[85] Kohl J, Neuman B and Ts'o T. The Evolution of the Kerveros Authentication Service. In Brazier F and Johansen D. Distributed Open Systems. Los Alamitos, CA: IEEE Computer Society Press, 1994, Available at http://web.mit.edu/kerberos/www/papers.html
[86] Lam K and Gollmann D. Freshness Assurance of Authentication Protocols. Proceedings, ESORICS'92, 1992, New York: Springer-Verlag
[87] Lam K and Beth T. Timely Authentication in Distributed Systems. Proceedings, ESORICS'92 1992, New York: Spring-Verlag
[88] Le A, Matyas S, Johnson D and Wilkins J. A Public Key Extension to the Common Cryptogrphic Architecture. IBM Systems Journal, No. 3, 1993
[89] Lewand R. Cryptological Mathematics. Washington, DC: Mathematical Association of America, 2000
[90] Lipmaa H, Rogaway P and Wagner D. CTR Mode Encryption. NIST First Modes of Operation Workshop, October 2000
http://csrc.nist.gov/encryption/modes

[91] Lodin S and Schuba C. Firewalls Fend Off Invasions from the Net. IEEE Spectrum, February 1998
[92] Macgregor R, Ezvan C, Liguori L and Han J. Secure Electronic Transactions: Credit Card Payment on the Web in Theory and Practice. IBM RedBook SG24-4978-00, 1997. Available at www. redbooks. ibm
[93] Madsen J. World Record in Password Checking. Usenet, comp. security. misc newsgroup, August 18, 1993
[94] Mantin I, Shamir A. A Practical Attack on Broadcast RC4. Proceedings, Fast Software Encryption, 2001
[95] Markham T. Internet Security Protocol. Dr. Dobb's Journal, June 1997
[96] Matsui M. Linear Cryptanalysis Method for DES Cipher. Proceedings, Eurocrypt'93, 1993, New York: Springer-Verlag
[97] Matyas S. Key Handling with Control Vectors. IBM Systems Journal, No. 2, 1991
[98] Matyas S, Le A and Abraham D. A Key-Management Scheme Based on Control Vectors. IBM Systems Journal, No. 2, 1991
[99] Meinel C. Code Red for the Web. Scientific American, October 2001
[100] Merkle R. Secrecy, Authentication, and Public Key Systems. Ph. D. Thesis, Standford University, June 1979
[101] Merkle R and Hellman M. On the Security of Multiple Encryption. Communications of the ACM, July 1981
[102] Merkle R. One Way Hash Functions and DES. Proceedings, CRYPTO'89, 1989, New York: Springer-Verlag
[103] Mayer C and Matyas S. Cryptography: A New Dimension in Computer Data Security, New York: Wiley, 1982
[104] Miller S, Neuman B, Schiller J and Saltzer J. Kerberos Authentication and Authorization System. Section E. 2. 1, Project Athena Technical Plan, M. I. T. Project Athena, Cambridge, MA. 27 October 1988
[105] Miller S. IPv6: The New Internet Protocol. Upper Saddle River, NJ: Prentice Hall, 1998
[106] Mitchell C, Walker M and Rush D. CCITT/ISO Standards for Secure Message Handling. IEEE Journal on Selected Areas in Communications, May 1989
[107] Mitchell C, Piper F and Wild P. Digital Signatures. In [SIMM92]
[108] Muftic S. Security Mechanisms for Computer Networks. New York: Ellis Horwood, 1989
[109] Nechvatal J. Public Key Cryptography. In[SIMM92]
[110] Nechvatal J, et al. Report on the Development of the Advanced Encryption Standard. National Institute of Standards and Technology. October 2, 2000
[111] Needham R and Schroeder M. Using Encryption for Authentication in Large Networks of Computers. Communications of the ACM, December 1978
[112] Neuman B and Stubblebine S. A Note on the Use of Timestamps as Nonces. Operating Systems Review, April 1993
[113] Oorschot P and Wiener M. A Known-Plaintext Attack on Two-Key Triple Encryption. Proceedings, EUROCRYPT'90, 1990, New York: Springer-Verlag
[114] Oppliger R. Internet Security: Firewalls and Beyond. Communications of the ACM, May 1997
[115] Pfleeger C. Security in Computing. Upper Saddle River, NJ: Prentice Hall, 1997
[116] Popek G and Kline C. Encryption and Secure Computer Networks. ACM Computing Surveys,

December 1979
[117] Rabin M. Digitalized Signatures In Foundations of Secure Computation. DeMillo R, Dobkin D, Jones A and Lipton R. eds. New York: Academic Press, 1978
[118] Rivest R, Shamir A and Adleman L. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Communications of the ACM, February 1978
[119] Rivest R. The MD4 Message Digest Algorithm. Proceedings, Crypto'90, August 1990, New York: Springer-Verlag
[120] Rivest R. The RC5 Encryption Algorithm. Proceedings, Second International Workshop on Fast Software Encryption, December 1994, New York: Springer-Verlag
[121] Schnorr C. Efficient Signatures for Smart Card. Journal of Cryptology, No. 3, 1991
[122] Schneier B. Secrets and Lies: Digital Security in a Networked World. New York: Wiley, 2000
[123] Shannon C. Communication Theory of Secrecy Systems. Bell Systems Technical Journal, No. 4, 1949
[124] Smith R. Internet Cryptography. Reading, MA: Addison-Wesley, 1997
[125] Stallings W. Data and Computer Communications. Sixth Edition. Upper Saddle River, NJ: Prentice Hall, 2000
[126] Steiner J, Neuman C and Schiller J. Kerberos: An Authentication Service for Open Networked Systems. Proceedings of the Winter 1988 USENIX Conference, February 1988
[127] Stevens W. TCP/IP Illustrated, Volume 1: The Protocols. Reading, MA: Addison-Wesley, 1994
[128] Thompson K. Reflections on Trusting Trust (Deliberate Software Bugs). Communications of the ACM, August 1984
[129] Tsudik G. Message Authentication with One-Way Hash Functions. Proceedings, INFOCOM'92, May 1992
[130] Tung B. Kerberos: A Network Authentication System. Reading, MA: Addison-Wesley, 1999
[131] Vaccaro H and Liepins G. Detection of Anomalous Computer Session Activity. Proceedings of the IEEE Symposium on Research in Security and Privacy, May 1989
[132] Voydock V and Kent S. Security Mechanisms in High-Level Network Protocols. Computing Surveys, June 1983
[133] Wack J, Cutler K and Pole J. Guidelines on Firewalls and Firewall Policy. NIST Special Publication SP 800-41, January 2002
[134] Williamson M. Thoughts on Cheaper Non-Secret Encryption. CESG Report, August 1976
[135] Woo T and Lam S. Authentication Revisted. Computer, April 1992
[136] Steiner M, Tsudik G, Waidrer M. Key agreement in dynamic peer groups[J]. IEEE Transactions on Parallel and Distributed Systems, 2000, 11(8): 769～780
[137] Michael B, Birgit P, Michael W. Cryptographic protocols/network security: a composable cryptographic library with nested operations[C]. Proceedings of the 10th ACMConference on Computer and Communications Security. New York: ACM Press, 2003: 220～230
[138] Byun J W, Lee D H. N-party encrypted Diffie-Hellman key exchange using different passwords [C]. ACNS 2005. LNCS: Berlin: Springer-Verlag, 2005, 3531: 75～90
[139] Dolev D, Yao A. On the security of public key protocols[J]. IEEE Transactions on Information Theory, 1983, 25(2): 198～208
[140] Aboba B, Simon D. On the Security of Public Key Protocols[J]. IEEE Trans on Information Theory, 1983, 29(2): 198～208
[141] Thomas S A. IPng and the TCP/IP Protocol: Implementing the Next Generation Internet. New

York:John Wiley & Sons,1996

[142] Canetti R. Universally Composable Security: a New Paradigm for Cryptographic Protocols[C]. Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science(FOCS). New York: IEEE Press,2001: 136 ～145

[143] Atkinson R J. Toward a More Secure Internet Computer. Jan. 1997,pp. 57～61

[144] Bellare M, Canetti R, Krawczyk H. A Modular Approach to the Design and Analysis of Authentication and Key-exchangeProtocols[C]. Proceedings of the 30th Annual Symp on the Theory of Computing. New York: ACM Press,1998: 419～428

[145] Oppliger R. Authentication Systems for Secure Networks. Norwood,Mass:Artech House,1996

[146] Canetti R,Krawczyk H. Security Analysis of IKE's Signature -based Key-exchange Protocol[C]. LNCS,2002,2442: 143～161

[147] Goldwasser S,Micali S,Rivest R. A Digital Signature Scheme Secure Against Adaptive Chosen-message Attacks[J]. SIAM Journal on Computing,1998,17(2): 281～308

[148] Bird R, Gobal I, Herzberg A and et al. Systematic design of a family of attack-resistant authentication protocols. Journal on Selected Areas in Communications,Vol. 11,No. 5,pp. 679～693,June 1993

[149] Gong L, Syverson P. Fail-stop protocols: An approach to designing secure protocols. Proc of Dependable Computing for Critical Applications 5, pp. 79 ～ 100, IEEE Computer Society Press,1998

[150] Kent S, Atkinson R. Security Architecture for the Internet Protocol. 1998

[151] Heather J, Lowe G and Schneider S. How to prevent type flaw attacks on security protocols. 13th Computer Security Foundations Workshop. IEEE Computer Society Press,2000

[152] Kent S and Atkinson R. Security architecture for the Internet protocol. IETF,RFC 2401,1998

[153] Guttman J. Security protocol design via authentication tests. 15th IEEE Computer Security Foundations Workshop,pp. 92～103,Cape Breton,Canada,IEEE Computer Society Press,2002

[154] Maughan D, Shertler M and Turner J. Internet Security Association and Key Management Protocol. Internet Standard Track. RFC 2408. November 1998

[155] Hongcai Tao,Dake He. Normal forms and normalization of authentication protocols. Proc of the 2006 International Conference on Computational Intelligence and Security, Vol. 2, pp. 1363～1366,Guangzhou,China,IEEE Press,2006

[156] Aziz A, Patterson M and Baehr G. Simple Key-Management for Internet Protocols (SKIP), Proc. 1995 Intel Networking Conf. ,Internet Soc. ,Reston, VA; available at http: //www. isoc. org/HMP/PAPER/244/abst. html

[157] Needham R,Schroeder M. Using encryption for authentication in large networks of computers. Communications of the ACM. 1978,21(12): 993～999

[158] Haller N The S/KEY One-Time Password System, Proc. Internet Soc. Symp. Network and Distributed System Security,Internet Soc. ,Reston, VA,1994,pp. 151～158

[159] Kailar R. Accountability in electronic commerce protocols. IEEE Transactions on Software Engineering. 1996,22(5): 313～328

[160] Secure Electronic Transaction (SET) Specification—Book 3: Formal Protocol Description, version 1. 0,Visa and MasterCard,May 1997; available online at http: //www-s2. visa. com/nt/ecomm/set/set_bk3. pdf

[161] Deng R H, Gong L. Practical protocols for certified electronic mail. Journal of Network and Systems Management. 1996,4(3): 279～297

[162] Abadi M and Needham R. Prudent engineering practice for cryptographic protocols, IEEE Transaction. Software. Eng. ,vol. 22,pp. 6～15,Jan. 1996
[163] Mudge and Schneier B. Cryptanalysis of Microsoft's point-to-point tunneling protocol (PPTP),In Proc. 5th ACM Conf. Computer and Communications Security,1998,pp. 132～141
[164] Georgiadis P,Gritzalis S and Spinellis D. Security protocols over open networks and distributed systems: Formal methods for their analysis,design and verification,Comput. Commun. ,vol. 22, pp. 695～707,May 1999
[165] Meadows C A. Formal verification of cryptographic protocols: A survey Advances in Cryptology—Asiacrypt'94 New York: Springer -Verlag,1995,pp. 133～150
[166] Clark S C, Freedman S B and Millen J K. The interrogator: Protocol security analysis, IEEE Trans. Softw. Eng. ,Vol. SE-13,No. 2,1987
[167] Dolev D and Yao A. On the security of public key protocols IEEE Trans. Inform. Theory,Vol. IT\-29,pp. 198～208,Mar. 1983
[168] Kemmerer R. Analyzing encryption protocols using formal verification techniques. IEEE J. Select. Areas Commun. ,Vol. 7,pp. 448～457,May 1989
[169] Snekkenes E. Formal Specification and Analysis of Cryptographic Protocols. Ph. D. dissertation Oslo,Norway: Univ. Oslo,1995
[170] Paulson L. Proving properties of security protocols by induction. In Proc. IEEE Security Foundations Workshop X,1997,pp. 70～83
[171] Paulson L. Mechanized proofs for a recursive authentication protocol. In Proc. IEEE Security Foundations Workshop X,1997,pp. 84～94
[172] Bolignano D. An approach to the formal verification of cryptographic protocols. In Proc. 3rd ACM Conf. Computer and Communications Security,1996,pp. 106～118
[173] Heintze N and Tygar J D. A model for secure protocols and their compositions. In Proc. IEEE Computer Society Symp. Research Security and Privacy,May 1994,pp. 2～13
[174] Bekmann J P, Goede P De and Hutchison A C M. SPEAR: Security protocol engineering and analysis resources. In Proc. DIMACS Workshop on Design and Formal Verification of Security Protocols,Sept. 1997
[175] Gong L. Efficient network authentication protocols: Lower bounds and optimal implementations. Distributed Comput. Vol. 9,pp. 131～145,1995
[176] Yi L J, Bai G Q, Xiao G Z. Proxy multi-signature scheme: A new type of proxy signature scheme. Electron Lett. 2000,36(6): 527～528
[177] Brackin S H and Lichota R W. CASE for high assurance: Utilizing commercial technology for automated cryptographic protocol analysis. In Proc. 6th Annual Dual-Use Technologies and Applications Conf. ,June 1996
[178] Shamir A. How to share a secret. Communication of ACM. 1979,22(11): 612～613
[179] Lichota R, Hammonds G and Brackin S H Verifying the correctness of cryptographic protocols using convince. In Proc. 12th IEEE Computer Security Applications Conf. ,1996,pp. 117～128
[180] Mambo M, Usuda K,Okamoto E. Proxy signatures: Delegation of the power to sign messages. IEICE Trans on Fundam. 1996,E79-A (9): 1338～1354
[181] Lichota R W, Hammonds G L and Brackin S H. Verifying cryptographic protocols for electronic commerce. Presented at the Proc. 2nd USENIX Workshop on Electronic Commerce Oakland, CA,Nov. 1996
[182] Sun H M. An efficient nonrepudiable threshold proxy signature scheme with known signers.

Computer Communications. 1999,22(8): 717～722
[183] Berry M S, Hutchison A C M and E. Saul. Predicting the performance of transactional electronic commerce protocols. In Proc. 7th Annual Working Conf. Information Security Management and Small Systems Security Amsterdam,The Netherlands,Sept. 1999,pp. 161～175
[184] Wang C T,Lin C H,Chang C C. Threshold signature schemes with traceable signers in group communications. Computer: Communications. 1998,21(8): 771～776
[185] Saul E and Hutchison A C M. A generic graphical specification environment for security protocol modeling. In Proc. 6th Annual Working Conf. Information Security Beijing,China,Aug. 2000, pp. 311～320
[186] Cao Zhenfu. A threshold key escrow scheme based on public key cryptosystem. Science in China (Series E). 2001,44(4): 441～448
[187] Saul E and Hutchison A C M. A graphical environment for the facilitation of logic-based security protocol analysis. South African Computer J. ,Vol. 26,pp. 196～200,Nov. 2000
[188] Tseng Y M,Jan J K. Attacks on threshold signature schemes with traceable signers . Information Processing Letters. 1999,71(1): 1～4
[189] Gong L. Handling infeasible specifications of cryptographic protocols. In Proc. 4th IEEE Security Foundations Workshop Franconia,NH,June 1991,pp. 99～102
[190] Gong L, Needham R and Yahalom R. Reasoning about belief in cryptographic protocols. In Proc. IEEE Symp. Research in Security and Privacy Oakland,CA,1990,pp. 234～248
[191] Caceres R and Iftode L. Improving the performance of reliable transport protocols in mobile computing environments. IEEE J. Select. Areas Commun. ,Vol. 13,pp. 850～857,June 1994
[192] Rescorla E. SSL and TLS: Designing and Building Secure Systems. Reading, MA: Addison-Wesley,2001
[193] Harkins D and Carrel D. The Internet key exchange (IKE). IETF,RFC 2409,1998
[194] Zhang Y and Singh B. A multi-layer IPSec protocol. In Proc. Usenix Security Symp. , Aug. 2000,pp. 213～228
[195] Abadi M, Needham R. Prudent engineering practice for cryptographic protocols. IEEE Transactions on Software Engineering,Vol. 22,No. 1,pp. 6～15,January 1996
[196] Abadi M. Explicit communication revisited: two new attacks on authentication protocols. IEEE Transactions on Software Engineering,Vol. 23,No. 1,pp. 185～186,March 1997
[197] Clark J A and Jacob J L. A survey of authentication protocol literature. Version 1. 0,November 1997,available at http: //www-users. cs. york. ac. uk/jac/
[198] Clark J A and Jacob J L. Protocols are Programs Too: the Meta-heuristic Search for Security Protocols. Information and Software Technology,Vol. 43,891～904,2001
[199] Heather J, Lowe G and Schneider S. How to prevent type flaw attacks on security protocols. In Proceedings of 13th IEEE Computer Security Foundations Workshop,255～268,2000
[200] Needham R and Schroeder M. Using encryption for authentication in large networks of computers. Comm. ACM,21(12),993～999,1978
[201] Neumann B C and Stubblebine S G. A note on the use of timestamps as nonces. ACM Operating Systems Reviews,27(2),10～14,April 1993
[202] Paulson L C. Proving security protocols correct. In Proceedings of 14th Symposium on Logic in Computer Science,370～381,1999
[203] Li Y,Yang W and Huang C W. Preventing Type Flaw Attacks On Security Protocols With A Simpli. Ed Tagging Scheme, Journal of Information Science and Engineering (accepted),

July 2004

[204] Yang W and Tsay C-W. A Logic Approach To The Veri. Cation And Testing Of Security Protocols, In Proceedings of International Conference on Communcations and Computer Networks (CCN 2002) (Cambridge,MA,November 4-6),140～145,2002

[205] Schulzrinne H and Rosenberg J. The Session Initiation Protocol: Internet-centric Signaling. IEEE Communications Magazine,Vol. 38,Oct 2001

[206] Camarillo G et al. Evaluation of transport protocols for the session initiation protocol. IEEE Network,Vol. 17,No. 5,Oct 2003

[207] Salsano S et al.. SIP security issues: the SIP authentication procedure and its processing load. IEEE Networks,Vol. 16,No. 6,Dec. 2002

[208] Rosenberg J et al. ,SIP: Session Initiation Protocol,IETF RFC 3261,Jun 2002

[209] SIP security modules for ns-2,available at http://hit. skku. edu/～eccha

[210] Karnin E D, Greene J W, Massey J L. On Secret Sharing Systems. IEEE Trans. Inform. Theory,1983,IT-29(1): 35～41

[211] Stinson D R. An Explication of Secret Sharing Schemes. DES. Codes Crypt, 1992, (2): 357～390

[212] Ito M Saito A and Nishizeki T. Secret Sharing Scheme Realizing General Access Structure. In Proceeding of IEEE Globecom'87. 99～102

[213] Benaloh J, Leichter J. Generalized secret sharing and monoton functions: advances in cryptology—CRPTO'88,Lecture Notes in Computer Science,Springer,Berlin,1990,27～35

[214] Abrams M D,Jajodia S,Podell H J. Information Security: An Integrated Collection Of Essays. California,USA: IEEE Computer Society Press Los Alamitos,1995

[215] James P Anderson. Computer Security Technology Planning Study. ESD -TR-73-51, ESD/AFSC,1972

[216] Atluri V, Jajodia S, Keefe T F, et al. Multilevel Secure Transaction Processing: Status and Prospects. Database Security Volume X, Status and Prospects, IFIP TCll/WG11. 3 Tenth Interntional Conference on Database Security: 79～98,1996

[217] Castano S,Fugini M G,Martella G,et al. Database Security. NY: ACM Press/Addison-Wesley Publishing Company,1995

[218] Fugini M G. Secure Database Development Methodologies. In Database Security: Status and Prospects. Annapolis,Maryland: North-Holland: 103～129,1988

[219] Spooner D L. Relationships Between Database System And Operating System Security. DBSec: 149～158,1987

[220] Boyd C and Mao W. Limitations Of Logical Analysis Of Cryptographic Protocols. EuroCrypto,1993

[221] Davies D W and Price W L. Security for Computer Nefworks (second edition). John Wiley & Sons,1989

[222] Gaarder K and Snekkenes E. Applying A Formal Analysis Technique To The CCITT X. 509 Strong Two-Way Authentication Protocol. Journal of Cryptology,3(2): 81～98,1991

[223] Kailar R and Gligor V D. On Belief Evolution In Authentication Protocols. In Proceedings of Computer Security Foundations Workshop,pp. 171～181. IEEE Computer Society Press,1991

[224] Otway D and Rees O. Efficient and timely mutual authentication. Operating Systems Review, Vol. 21(1): 8～10,1987

[225] Syverson P. The Use Of Logic In The Analysis Of Cryptographic Protocols. In Proceedings of

Computer Security Foundations Workshop, pp 156～170. IEEE Computer Society Press, 1991
[226] Abadi M and Needham R. Prudent Engineering Practice For Cryptographic Protocols. Research Report 125, Digital Equipment Corporation Systems Research Center, 1994
[227] Bird R, Gopal I, Herzberg A, Janson P A, Kutten S, Mulva R and Yung M. Systematic Design Of Two-Party Authentication Protocols. In Proceedings of Crypt0'91: Advances in Cryptology, volume 576 of Lecture Notes in Computer Science, pp. 44～61. Springer-Verlag, 1991
[228] Clark J and Jacob J. On the security of recent protocols. Information Processing Letters, 56(3): 151～155, 1995
[229] Diffie W, Van Oorschot P C and Wiener M J. Authentication And Authenticated Key Exchanges. Designs, Codes and Cryptography, 2: 107～125, 1992
[230] Hwang T and Chen Y-H. On The Security Of SPLICE/AS—The Authentication System In WLDE Internet. Information Processing Letters, 53: 97～101, 1995
[231] Hwang T, Lee N-Y, Li C-M, KO M-Y and Chen Y-H. Two attacks on Neuman-Stubblebine authentication protocols. Information Processing Letters, 53: 103～107, 1995
[232] Lowe G. An attack on the Needham-Schroeder public-key authentication protocol. Information Processing Letters, 56: 131～133, 1995
[233] Millenl J K, Clark S C and Freedman S B. The interrogator: Protocol security analysis IEEE Transactions on software Engineering, 13(2), 1987
[234] Needham R and Schroeder M. Using encryption for authentication in large networks of computers. Communications of the ACM, 21(12): 993～999, 1978
[235] Roscoe A W. Modelling and verifying key-exchange protocols using CSP and FDR. In 8th IEEE Computer Security Foundations Workshop, 1995
[236] Roscoe A W. Intensional spxifications of security protocols. In 9th IEEE Computer Security Foundations Workshop, 1996
[237] Snekkenes E. Roles in cryptographic protocols. In Proceedings of IEEE Symposium on Research in Security and Privacy, pp. 105～119, 1992
[238] Syverson P. On key distribution protocols for repeated authentication. Operating Systems Review, 27(4): 24～30, 1993
[239] Woo T Y C and Lam S S. A lesson on authenticated protocol design. Operating Systems Review, 28(3): 24～37, 1994
[240] Yamaguchi S, Okayama K and Miyahara H. Design and implementation of an authentication system in WIDE Internet environment. In Proc. 10th IEEE Region Conf. on Computer and Communication Systems, 1990
[241] Roscoe A W. Proving security protocols with model checkers by data independence techniques. In 11th Computer Security Foundations Workshop, pp. 84～95, 1998
[242] The SSL protocol. Available via http://home.netscape.com/newsref/std/SSL.html, 1996
[243] Misarsky J. How (Not) to Design RSA Signature Schemes, Public Key Cryptography. Imai H and Zheng Y(Eds.). Lecture Notes in Computer Science 1431, Springer-Verlag, 1998
[244] Preneel B, Rijmen V and Bosselaers A. Recent Developments in the Design of Conventional Cryptographic Algorithms, State of the Art in Applied Cryptography. Preneel B and Rijmen V (Eds.). Lecture Notes in Computer Science 1528, Springer-Verlag, 1998
[245] Brackin S. Automatically Detecting Most Vunerabilities in Cryptographic Protocols, in The DARPA Information Survivability Conf and Exposition. Jan 2000, Vol. 1, pp. 222～236
[246] Justin Childs. Evaluating the TLS Family of Protocols with Weakest Precondition Reasoning.

Technical Report. Florida State University Department of Computer Science. http://www.cs.fsu.edu/research/reports

[247] Dierks T and Allen C. The TLS protocol: Version 1.0. Request for Comments: 2246, available at ftp://ftp.isi.edu/in-notes/rfc2246.txt

[248] Kelsey J, Schneier B and Wagner D. Protocol interactions and the chosen protocol attack, in Lecture Notes in Computer Science, 1361. pp. 91～104, Heidelberg: Springer, Berlin, 1998

[249] Catherine Meadows. Analysis of the Internet Key Exchange Protocol Using the NRL Protocol Analyzer. IEEE Symposium on Security and Privacy, 1999

[250] Mitchell J C, Shmatikov V and Stern U. Finite-state analysis of SSL 3.0 and related protocols, In Workshop on Design and Formal Verification of Security Protocols Sept. 1997, DIMACS

[251] Paulson L C. Inductive Analysis of the Internet Protocol TLS Technical Report 440. Cambridge University Computer Science Laboratory, 1998

[252] Wagner D and Schneier B. Analysis of the SSL 3.0 Protocol, In D. Tygar Ed., USENIX Workshop on Electronic Commerce, pp. 29～40. USENIX Association, 1996

[253] Alec Yasinsac and William A Wulf. A Framework for A Cryptographic Protocol Evaluation Workbench Proceedings of the Fourth IEEE International High Assurance Systems Engineering Symposium (HASE99). Washington D.C., Nov. 1999

[254] Peluso L, Cotroneo D, Romano S P, Ventre G ASSYST: an Active Security System against DoS attacks-Technical Report. Apr. 2001, Dept. of Computer Sciences, University of Napoli, Italy

[255] W W W Consortium. The World Wide Web Security FAQ. http://www.w3.org/Security/faq/www-security-faq.html, 1998

[256] IPSec Working Group. http://www.ietf.org/html.charters/ipsec-charter.html

[257] Burke J, McDonald J and Austin T. Architectural support for fast symmetric-key cryptography. In Proc. Int. Conf. ASPLOS, pp. 178～189, Nov. 2000

[258] Potlapally N, Ravi S, Raghunathan A and Lakshminarayana G. Optimizing public-key encryption for wireless clients. In PMC. IEEE Int. Conf. Communications, pp. 1050～1056, May 2002

[259] Apostolopoulos G, Peris V, Pradhan P and Saha D. Securing electronic commerce: Reducing the SSL overhead. IEEE Network, pp. 8～16. July 2000

[260] Wong D S and Chan A H. Mutual authentication and key exchange for low power wireless communications. In Proc. IEEE MILCOM Conf. pp. 39～43, Oct. 2001

[261] Open SSL Project. http://www.openssl.org

[262] Yin Y L. The RC5 encryption algorithm: Two years on RSA Laboratories' Cryptobytes. Vol. winter, pp. 14～15, 1997

[263] Miltchev S, Loamidis S and Kcromytis A D. A Study of the Relative Costs of Network Security Protocols, in Proceedings of the USENIX Annul Technical Conference. Freenix Track, pp. 41～48, June 2002

[264] Boneh D and Shacham N. Improving SSL Handshake Performance via Batching, in Proceedings of the RSA Conference, January 2001

[265] Goldberg A, Buff R and Schmin A. Secure Web Server Performance Dramatically Improved By Caching SSL Session Keys in Workshop on Internet Server Performance. Held in conjunction with SIGMETRICS, June 1998

[266] Boe M (1998). TLS-based Telnet Security. [cited 15 Dec 1998] http://www.ietf.org/internetdrafts/drafts-ietf-tn3270e-telnet-tls-01.txt

[267] Stallings W. Network Security Essentials: Applications and Standards. New Jersey: Prentice

Hall,2000

[268] Wood Y K. Transport Layer Security (TLS) using RSA and RC4 Cryptography Modules, University of Malaya (unpublished),1999

[269] Thomas A S. SSL and TLS Essentials: Securing the Web. New York: Wiley,2000

[270] Roscarla E. SSL and TLS: Designing and Building Secure Systems. Boston: Addison-Wesley,2001

[271] Asokan N, Schunter M, Waidner M. Optimistic Fair Exchange of Digital Signatures. IEEE Journal on Selected Areas in Communications 18,2000,pp. 593~610

[272] Ateniese G. Efficient Verifiable Encryption (and Fair Exchange) of Digital Signatures. Proceedings of ACM Conference on Computer and Communications Security,1999,pp. 138~146

[273] Bao F, Deng R, Mao W. Efficient and Practical Fair Exchange Protocols with Off-line TTP. Proceedings of IEEE Symposium on Security and Privacy,1998,pp. 77~85

[274] Chen L. Efficient Fair Exchange with Verifiable Confirmation of Signatures. Proceedings of Advances in Cryptology—ASIACRYPT'98,LNCS, Vol. 1514,Springer-Verlag,Berlin,Germany, 1998,pp. 286~299

[275] Deng R H, Gong L, Lazar A A, Wang W. Practical Protocols for Certified Electronic Mail. Journal of Network and System Management,4(3),1996,pp. 279~297

[276] Markowitch O, Saeednia S. Optimistic Fair Exchange with Transparent Signature Recovery. Proceedings of 5th International Conference Financial Cryptography 2001, LNCS, Springer-Verlag,2001

[277] Ray I. An Optimistic Fair Exchange Ecommerce Protocol with Automated Dispute Resolution. Proceedings of 1st International Conference on E-Commerce and Web Technologies EC-Web 2000,LNCS, Vol. 1875,Springer-Verlag,Berlin,Germany,2000,pp. 84~93

[278] Shi Q, Zhang N, Merabti M. Signature-based approach to fair document exchange. Communications,IEEE Proceedings,150(1),2003,pp. 21~27

[279] Zhou J, Deng R, Bao F. Some Remarks on a Fair Exchange Protocol, Proceedings of International Workshop on Practice and Theory in Public Key Cryptography. LNCS, Vol. 1751, Springer-Verlag,2000,pp. 46~57

[280] Adi K and Debbabi M. A game semantics and approach for security procotols. In 6th International Symposium on Programming and Systems,ISPS'2003,pp 209~227,2003

[281] Adi K, Debbabi M and Mejri M. A New Logic for Electronic Commerce Protocols. Theoretical Computer Science (TCS),291(3),2003

[282] Lowe G. SPLICE-AS, A Case Study in Using CSP to Detect Errors in Security Protocols. Technical report,Programming Research Group,Oxford,1996

[283] Stoller S D. A bound on attacks on payment protocols. In Logic in Computer Science,pp. 61~70,2001

[284] Lee H and Kim K. An Adaptive Authentication Protocol based on Reputation for Peer-to-Peer System. The 2003 Symposium on Cryptography and Information Security,Jan 2003

[285] Leslie Lamport. Password Authentication with Communication. Communications of the ACM, pp. 770~771,Number 81,Volume 24,November 1981

[286] Adrian Perrig,Ran Canetti,Tygar J D and Dawn Song. The TESLA Broadcast Authentication Protocol. RSA Cryptobytes,Summer 2002

[287] Wong D S,Chan A H. Efficient and mutually authenticated key exchange for low power computing devices. Proceedings of ASIACRYPT,2001

[288] Beller M J, Chang L F, Yacobi Y. Privacy and authentication on a portable communications system. IEEE Journal on Selected Areas in Communications, 1993, pp. 16～23
[289] Bella G, Massacci F and Paulson L C. Verifying the SET registration protocols. IEEE Journal of Selected Areas in Communications, 21(1): 77～87, 2003
[290] Bellare M, Garay J A, Hauser R, Herzberg A, Krawczyk H, Steiner M, Tsudik G, Herreweghen E V, Waidner M. Design, implementation and deployment of the Ikp secure electronic payment system. IEEE Journal of Selected Areas in Communications 18, 2000, pp. 611～627
[291] Cox B, Tygar J D and Sirbu M. NetBill Security and Transaction Protocol. In Proceedings of the First USENIX Workshop in Electronic Commerce, July, 1995, pp. 77～88
[292] Fielding R, Gettys J, Mogul J, Frystyk H, Masinter L, Leach P and Berners-Lee T. Hyper Text Transfer Protocol-HTTP/1. 1. Standards Track, RFC 2616, Internet Engineering Task Force, 1999
[293] IP Encapsulating Security Payload (ESP). Standards Track, RFC 2406, Internet Engineering Task Force, 1998
[294] Postel J. Transmission Control Protocol. RFC 793, USC/Information Services Institute, 1981
[295] Zhang Y. A multilayer IP security protocol for TCP performance enhancement in wireless networks. IEEE Journal on Selected Areas in Communications, May 2004
[296] Rabin M O. Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance. Journal of ACM, Vol. 36, No. 2, pp. 335～348, Apr. 1989
[297] Frankel Y. A Practical Protocol for Large Group Oriented Networks Proc. Of Eurocrypt 1989
[298] Perrig A, Szewczyk R, Wen V, Cullar D and Tygar J D. Spins: Security Protocol for Sensor Networks. In Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Rome, Italy, July 2001, pp. 189～199
[299] Batina L, Ors S B, Preneel B, Vandewalle J. Hardware Architectures for Public Key Cryptography. Integration, The VLSI Journal, 34 (2003), pp. 1～64
[300] Abadi M. Secrecy by typing in security protocols. Journal of the ACM, Vol. 46, pp. 749～786, September 1999
[301] Clarke E, Jha S and Marrero W. Partial order reductions for security protocol verification. Tools and Algorithms for the Construction and Analysis of Systems, 2000
[302] Cohen E. TAPS: a first-order verifier for cryptographic protocols. Proceedings of the 13th IEEE Computer Security Foundations Workshop, pp. 144 ～ 158. IEEE Computer Society Press, June 2000
[303] Dolev D, Even S and Karp R. On the security of Ping-Pong protocols. Information and Control, pp. 57～68, 1982
[304] Even S and Goldreich O. On the security of multi-party ping-pong protocols. In Proceedings of the 24th IEEE Symposium on the Foundations of Computer Science, pp. 3439, IEEE Computer Society Press, 1983
[305] Longley D and Rigby S. An automatic search for security flaws in key management schemes. Computers and Security, Vol. 11, pp. 75～90, 1992
[306] Lowe G. Towards a completeness results for model checking security protocols. Journal of Computer Security, Vol. 7, pp. 89～146, 1999
[307] Meadows C. Applying Formal Methods to the Analysis of a Key Management Protocol. Journal of Computer Security, Vol. 1, pp. 5～53, 1992
[308] Song D, Berezin S and Perrig A. Athena: A Novel Approach To Efficient Automatic Security

Protocol Analysis. Journal of Computer Security, Vol. 9, pp. 47～74, 2001

[309] Thayer F Fabrega, Herzog J and Guttman J. Strand spaces: Why is a security protocol correct? In Proceedings of the 1998 IEEE Symposium on Security and Privacy, pp. 160～171. IEEE Computer Society Press, May 1998

[310] Perrig A and Song D. A first step towards the automatic generation of security protocols. In Network and Distributed System Security Symposium, NDSS'00, pp. 73～84, February 2000

[311] Bhargavan K, Fournet C and Gordon A D. Verifying policy-based security for web services. In 11th ACM Conference on Computer and Communications Security (CCS'04), pp. 268～277, Oct. 2004

[312] Bhargavan K, Fournet C and Gordon A D. Policy advisor for WSE 3.0. In Web Service Security: Scenarios, patterns, and implementation guidance for Web Services Enhancements (WSE) 3.0. Microsoft Press, 2006

[313] Bhargavan K, Fournet C, Gordon A D and Pucella R. TulaFale: A security tool for web services. In International Symposium on Formal Methods for Components and Objects (FMCO'03), volume 3188 of LNCS, pp. 197～222, Springer, 2004

[314] Perrig A, Song D and Phan D. AGVI—automatic generation, verification, and implementation of security protocols. In 13th Conference on Computer Aided Verification (CAV), LNCS, pp. 241～245, Springer, 2001

[315] Murphy S, Lewis E, Puga R, Watson R. Strong Security for Active Networks. IEEE OpenArch 2001

[316] Cheng L, Galis A. Strong Authentication for Active Networks. IEEE-Softcom 2003, http://www.ee.ucl.ac.uk/～lcheng/Papers/SOFTCOM_2003.pdf

[317] Murphy S, Hayatnagarkar A, Krishnaswamy S. Prophylactic, Treatment and Containment Techniques for Ensuring Active Network Security, IEEE DARPA, 2003

[318] Doraswamy N, Harkins D. IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks, 2nd edition, ISBN: 0-13-046189-X, Prentice-Hall PTR, 2003, pp. 220～232

[319] Aiello W, Bellovin S, Blaze M, Canetti R, Ioannidis J, Keromytis A, Reingold O. Efficient, DoS-Resistant Secure Key Exchange for Internet Protocols. ACM Computers and Communications Security Conference(CCS), 2002

[320] Kam P, Simpson W. The Photuris Session Key Management Protocol. draft-ietf-ipsec-photuris-03.txt, September 1995

[321] Eronen P. Denial of service in public key protocols. In Proceedings of the Helsinki University of Technology Seminar on Network Security, December 2000, http://www.niksula.hut.fi/～peronen/publications/netsec_2000.pdf

[322] Matsuura K, Imai H. Modification of Internet Key Exchange Resistant against Denial-of-Service. In Proc. Of Internet Workshop 2000(IWS'2000), pp. 167～174, Feb. 2000

[323] Coppersmith D, Franklin M, Patarin J, Reiter M. Low-exponent RSA with related message. In Advances in Cryptology—EUROCRYPT 1996, Vol. 1070 of LNCS, Springer-Verlag, pp. 1～9

[324] Cheng L, Galis A. Simple Key Exchange for Active Networks, IEEE-ICON 2005, http://www.ee.ucl.ac.uk/～lcheng/Papers/ICON_2005.pdf

[325] Steiner M, Tsudik G and Waidner M. Cligues: A New Approach to Group Key Agreement. Proc. 18th Int'l Conf. Distributed Computing Systems (ICDCS'98). IEEE CS Press, 1998, pp. 380～387

[326] Nguyen L H and Roscoe A W. Efficient Group Authentication Protocols Based on Human Interaction. Proc. FCS-ARSPA,2006,pp. 9～32
[327] Diffie W and Hellman M E. New Directions in Cryptography. IEEE Trans. Information Theory,Vol. 22,No. 6,1976,pp. 644～654
[328] Laur S and Nyberg K. Efficient Mutual Data Authentication Using Manually Authenticated Strings. Proc. 5th Int'l Conf. Cryptology and Network Security (CANS'06), LNCS 4301, Springer,2006,pp. 90～107
[329] Ku W C and Chen S M. Weakness and improvements of an efficient password based remote user authentication scheme using smart card. IEEE Trans. On Consumer Electron. ,Vol. 50,No. 1, pp. 204～207,2004
[330] Sun H, Chen B and Hwang T. Secure key agreement protocols for three-party against guessing attacks. The Journal of Systems and Software,Vol. 5,No. 12,pp. 497～499,2001
[331] Enge A. Elliptic Curves and Their Applications to Cryptography. Norwell, MA: Kluwer Academic Publishers,1999
[332] Fernandes A. Elliptic Curve Cryptography. Dr. Dobb's Journal,December 1999
[333] Jurisic A and Menezes A. Elliptic Curves and Cryptography. Dr. Dobb's Journal,April 1997
[334] Guajardo J, Paar C. Efficient algorithms for elliptic curve cryptosystems. Advances in Cryptology-Crypto'99 of LNCS,1999,(1716): 75～85
[335] 美国国家安全局著.中国国家 973 信息与网络安全体系研究课题组组织翻译. 信息保障技术框架 3.0. 北京：中软电子出版社,2002
[336] (美)Bruce Schneier 著.吴世忠等译. 应用密码学——协议、算法与 C 语言程序. 北京：机械工业出版社,2000
[337] 中华人民共和国国家标准,GB17859-1999.计算机信息系统安全保护登记划分准则
[338] 冯登国. 密码分析学. 北京：清华大学出版社,桂林：广西科学技术出版社,2000
[339] 冯登国,蔡吉人.网络安全与密码学.贵州：贵州科技出版社,2004
[340] Bauer F L.密码编码和密码分析——原理与方法.北京：机械工业出版社,2001
[341] 龙毅宏.国外 PKI/CA 体系发展状况的研究,计算机安全,2001,09
[342] 王可.MD5 算法研究,中文信息,2002,02
[343] 龙冬阳.应用编码与计算机密码学.北京：清华大学出版社,2005
[344] 李大友主编,戴英侠,许剑卓等编著.计算机网络安全.北京：清华大学出版社,2005
[345] 何德全.安全协议.北京：清华大学出版社,2005
[346] 李海泉,李健.计算机网络安全与加密技术.北京：科学出版社,2001
[347] 张敏,冯登国.数据库安全.北京：科学出版社,2005
[348] 楚狂等.网络安全与防火墙技术.北京：人民邮电出版社,2000
[349] (美)Keith E Strassberg,Richard J Gondek,Gary Rollie 等著.李昂,刘芳萍,杨旭,程鹏等译.防火墙技术大全.北京：机械工业出版社,2003
[350] 张福泰,李继国等主编.密码学教程.武汉：武汉大学出版社,2006
[351] Abadi M,Gordon A D. A calculus for cryptographic protocols: The spi calculus. In 4th ACM Conference on Computer and Communications Security,1997,36～47

# 教师反馈表

感谢您购买本书！清华大学出版社计算机与信息分社专心致力于为广大院校电子信息类及相关专业师生提供优质的教学用书及辅助教学资源。

我们十分重视对广大教师的服务，如果您确认将本书作为指定教材，请您务必填好以下表格并经系主任签字盖章后寄回我们的联系地址，我们将免费向您提供有关本书的其他教学资源。

| | | | |
|---|---|---|---|
| 您需要教辅的教材： | | | |
| 您的姓名： | | | |
| 院系： | | | |
| 院/校： | | | |
| 您所教的课程名称： | | | |
| 学生人数/所在年级： | ______人/　1　2　3　4　硕士　博士 | | |
| 学时/学期 | ______学时/______学期 | | |
| 您目前采用的教材： | 作者：______<br>书名：______<br>出版社：______ | | |
| 您准备何时用此书授课： | | | |
| 通信地址： | | | |
| 邮政编码： | | 联系电话 | |
| E-mail： | | | |
| 您对本书的意见/建议： | | 系主任签字<br><br>盖章 | |

我们的联系地址：

清华大学出版社　学研大厦 A602，A604 室

邮编：100084

Tel：010-62770175-4409，3208

Fax：010-62770278

E-mail：liuli@tup. tsinghua. edu. cn；hanbh@tup. tsinghua. edu. cn